```
                        0001  *H RML 380Z MONITOR V 2.3 R
                        0002
                        0003  ;-----------------------------------------------------------
                        0004  ;************ R.M.L. CASSETTE OPERATING SYSTEM ***********
                        0005  ;************  MONITOR VERSION 2.3 R  REV. 0   ***********
                        0006  ;-----------------------------------------------------------
                        0007  ;-----------------------------------------------------------
                        0008
                        0009  ;       "BY VIEWING NATURE, NATURE'S HANDMAID, ART,
                        0010  ;        MAKES MIGHTY THINGS FROM SMALL BEGINNINGS GROW."
                        0011  ;                                          DRYDEN
                        0012
0000                    0013          ORG     0
      1000              0014  NXTROM  EQU     1000H
      4000              0015  RAM     EQU     4000H
                        0016
0000  C33703            0017  MONST:  JP      BEGIN           ;RST 00H (& POWER ON RESET)
0003  C37103            0018          JP      CONTC           ;ADDR FOR REENTER CMD
0006  544F              0019  MSGM:   DEFM    'TO'
0008  C32E40            0020          JP      RST8            ;RST 08H
000B  42524541          0021  MSGB:   DEFM    'BREAK'
0010  C33140            0022          JP      RST10           ;RST 10H
0013  3F455252          0023  MSGE:   DEFM    '?ERR?'
0018  C33440            0024          JP      RST18           ;RST 18H
001B  4649525?          0025  MSGF:   DEFM    'FIRST'
0020  C33740            0026          JP      RST20           ;RST 20H
0023  5354415?          0027  MSGS:   DEFM    'START'
0028  C38500            0028          JP      RCALL           ;RST 28H (RELATIVE CALL)
002B  4C41535?          0029  MSGL:   DEFM    'LAST'
002F  FF                0030          DEFB    -1
0030  1817              0031          JR      TRAP            ;RST 30H (TRAP CALL)
                        0032
                        0033  ;32H - THE NEXT TWO BYTES OF CODE ENABLE USER PROGRAMS TO
                        0034  ;FIND OUT WHERE THEY HAVE BEEN LOADED, MAKING IT POSSIBLE TO
                        0035  ;WRITE POSITION INDEPENDENT CODE (PIC).  THEY SHOULD BE
                        0036  ;CALLED ('CALL 32H'); ON RETURN OLD HL IS ON THE STACK
                        0037  ;AND HL CONTAINS THE ADDR OF THE CURRENT INSTRUCTION
                        0038  ;I.E. THE ONE AFTER THE CALL.
                        0039
0032  E3                0040          EX      (SP),HL
0033  E9                0041          JP      (HL)
                        0042
0034  57495448          0043  MSGF:   DEFM    'WITH'
                        0044
                        0045  ;BREAKPOINT - COME HERE AFTER RST 38H (0FFH)
                        0046  ;(INTERRUPTS IN MODE 1 ALSO COME HERE)
                        0047  ;OLD PC ON THE STACK IS DECREMENTED, TO POINT
                        0048  ;TO THE LOCATION OF THE BREAKPOINT AND THIS ADDRESS
                        0049  ;IS ALSO PUT IN (MPTR)
                        0050
0038  E3                0051  BREAK:  EX      (SP),HL         ;HL <- ADDR OF NEXT INSTR
0039  2B                0052          DEC     HL              ;POINT TO BREAK POINT
003A  220740            0053          LD      (MPTR),HL
003D  E3                0054          EX      (SP),HL
```

```
003E  E5        0055          PUSH    HL
003F  F5        0056          PUSH    AF
0040  210B00    0057          LD      HL,MSGB         ;ISSUE BREAK MESSAGE
0043  CDCC05    0058          CALL    MOUT
0046  C3D600    0059          JP      FPM             ;GO TO FP WITH HL/AF ON STACK
                0060
                0061  ;TRAP - A TRAP INSTRUCTION ('EMT'), SIMULATED BY RST 30H,
                0062  ;IS A TWO BYTE INSTRUCTION EQUIVALENT TO 'CALL'.  THE SECOND
                0063  ;BYTE CONTAINS THE TRAP CODE OF THE DESIRED TARGET ROUTINE,
                0064  ;AS FOLLOWS:
                0065  ;
                0066  ;          CODE            TARGET
                0067  ;
                0068  ;           -VE            ?ERR?
                0069  ;            0             MONITOR RESTART
                0070  ;          1-10            TRANSFER VECTORS ('VTV' - 'IN3')
                0071  ;         11-24            MONITOR ROUTINES IN 'TRTBL'
                0072  ;         25-127           JUMP THROUGH 'TRAPX'
                0073
0049  E5        0074  TRAP:   PUSH    HL              ;SAVE HL AND AF
004A  F5        0075          PUSH    AF
004B  CD9200    0076          CALL    LDRET           ;HL <- RET ADDR, INC RET ADDR
004E  7E        0077          LD      A,(HL)          ;A <- TRAP CODE
004F  B7        0078          OR      A
0050  28B1      0079          JR      Z,MONST+3       ;TRAP 0 - RESTART
0052  FAF305    0080          JP      M,ERROR         ;-VE TRAP CODE MEANS ERROR
                0081
                0082  ;IF CODE IS > 24(10), JUMP THROUGH TRAP EXTENSION VECTOR 'TRAPX'
                0083  ;WITH THE CODE IN REG A AND AF, HL AND THE RET ADDR ON THE STACK.
                0084
0055  FE19      0085          CP      25
0057  D23D40    0086          JP      NC,TRAPX
005A  D60B      0087          SUB     11
005C  381C      0088          JR      C,TR2
                0089
                0090  ;CODES 11 TO 24 (DECIMAL, INCLUSIVE) CALL THE MONITOR
                0091  ;ROUTINES WHOSE ADDRESSES ARE IN THE JUMP TABLE 'TRTBL'
                0092
005E  87        0093          ADD     A,A             ;A <- A * 2
005F  21B507    0094          LD      HL,TRTBL        ;HL <- ADDR OF DISPATCH TABLE
0062  85        0095          ADD     A,L             ;HL <- ADDR OF ADDR
0063  6F        0096          LD      L,A
0064  1802      0097          JR      TR1             ;AVOID NMI
                0098
0066  183A      0099          JR      NMI             ;LOCATION 66H
                0100
0068  7E        0101  TR1:    LD      A,(HL)          ;HL <- DISPATCH ADDR
0069  23        0102          INC     HL
006A  66        0103          LD      H,(HL)
006B  6F        0104          LD      L,A
006C  1814      0105          JR      TR3
                0106
                0107  ;.CHAN (EMT CHAN) ALLOWS CHANNEL TO BE PASSED IN REGISTER C
                0108
```

```
006E  E5        0109  .CHAN:  PUSH    HL
006F  F5        0110          PUSH    AF
0070  79        0111          LD      A,C             ;A <- CHANNEL
0071  FE01      0112          CP      1               ;CHECK THAT IT IS IN THE
0073  3001      0113          JR      NC,.CHN2        ; RANGE 1 TO 10
0075  FF        0114  .CHN1:  RST     38H             ;ELSE BREAK
0076  D60B      0115  .CHN2:  SUB     11
0078  30FB      0116          JR      NC,.CHN1
                0117
                0118  ;CODES 1 TO 10 (DECIMAL, INCLUSIVE) REACH THE TRANSFER VECTORS
                0119
007A  6F        0120  TR2:    LD      L,A             ;A <- A * 3
007B  87        0121          ADD     A,A
007C  85        0122          ADD     A,L
007D  212E40    0123          LD      HL,IN3+3        ;CALCULATE VECTOR ADDR
0080  85        0124          ADD     A,L
0081  6F        0125          LD      L,A
0082  F1        0126  TR3:    POP     AF              ;RESTORE AF AND HL
0083  E3        0127          EX      (SP),HL
0084  C9        0128          RET                     ;GO TO ADDR
                0129
                0130  ;RCALL - CALL RELATIVE ('CALR') IS A TWO BYTE PSEUDO-
                0131  ;INSTRUCTION SIMULATED BY RST 28H.  THE SECOND BYTE
                0132  ;CONTAINS THE OFFSET TO THE TARGET ADDR WITH THE SAME
                0133  ;CONVENTION AS THE RELATIVE JUMPS (I.E. TARGET-$-1).
                0134  ;THIS CODE CONVERTS THE OFFSET TO AN ABSOLUTE ADDR, THEN
                0135  ;JUMPS THERE, WITH THE UPDATED RETURN ADDR ON THE STACK.
                0136
0085  E5        0137  RCALL:  PUSH    HL              ;SAVE HL AND AF
0086  F5        0138          PUSH    AF
0087  CD9200    0139          CALL    LDRET           ;HL <- ADDR OF OFFSET
008A  D5        0140          PUSH    DE              ;SAVE DE
008B  CD6502    0141          CALL    RTOA            ;HL <- HL + (HL) + 1
008E  D1        0142          POP     DE              ;RESTORE DE, AF AND HL
008F  F1        0143          POP     AF
0090  E3        0144          EX      (SP),HL
0091  C9        0145          RET                     ;GO TO CALL
                0146
                0147  ;LDRET - LOAD HL WITH PREVIOUS RETURN ADDR, INCREMENT
                0148  ;RETURN ADDR.  ASSUMES HL AND AF ARE ALREADY ON THE STACK.
                0149  ;DESTROYS CY FLAG
                0150
0092  D5        0151  LDRET:  PUSH    DE
0093  210800    0152          LD      HL,8            ;HL <- POINTER TO STACK
0096  39        0153          ADD     HL,SP
0097  5E        0154          LD      E,(HL)          ;DE <- RETURN ADDR
0098  23        0155          INC     HL
0099  56        0156          LD      D,(HL)
009A  13        0157          INC     DE              ;INC RETURN AND STORE
009B  72        0158          LD      (HL),D
009C  2B        0159          DEC     HL
009D  73        0160          LD      (HL),E
009E  1B        0161          DEC     DE              ;CORRECT ADDR
009F  EB        0162          EX      DE,HL           ;HL <- ADDR
```

```
00A0  D1        0163          POP     DE
00A1  C9        0164          RET
                0165
                0166  ;NMI - COME HERE AFTER NMI CAUSED BY RESET BUTTON OR
                0167  ;SINGLE STEP LOGIC.
                0168  ;IF RESET, WAIT FOR BUTTON RELEASE THEN RESTART MONITOR
                0169  ;ELSE DISABLE SINGLE STEP BIT IN 'MASK' AND CHECK NMI FLAG:
                0170  ;IF (SSFLG) = 'R', JUMP TO USER PROG THROUGH NMIX,
                0171  ;ELSE IF (SSFLG) = 'S' (MONITOR SINGLE STEP) GO TO FP
                0172  ;ELSE TREAT AS RESET BUTTON GLITCH AND RESTART MONITOR
                0173
00A2  E5        0174  NMI:    PUSH    HL
00A3  F5        0175          PUSH    AF
00A4  21FCCF    0176          LD      HL,PORT0        ;DISABLE SINGLE STEP IF SET
00A7  3692      0177          LD      (HL),MKINIT
00A9  23        0178          INC     HL              ;POINT TO PORT 1
00AA  23        0179          INC     HL
00AB  CB56      0180          BIT     RESET,(HL)      ;WAS IT RESET BUTTON?
00AD  2009      0181          JR      NZ,NMI2         ;IF NOT
                0182
00AF  CB56      0183  NMI1:   BIT     RESET,(HL)      ;ELSE WAIT FOR BUTTON RELEASE
00B1  28FC      0184          JR      Z,NMI1
00B3  AF        0185          XOR     A               ;MAKE SURE SS FLG IS CLEAR
00B4  320440    0186          LD      (SSFLG),A
00B7  C7        0187  NMI1A:  RST     0               ;GO TO POWER ON ENTRY
                0188
00B8  210340    0189  NMI2:   LD      HL,MASK         ;NOT RESET BUTTON,
00BB  CBCE      0190          SET     SINGLE,(HL)     ;(CLEAR SINGLE STEP ANYWAY)
00BD  23        0191          INC     HL              ;POINT TO SS FLG
00BE  7E        0192          LD      A,(HL)          ;TEST FLAG
00BF  3600      0193          LD      (HL),0          ;THEN CLEAR IT
00C1  D652      0194          SUB     'R'
00C3  CA3A40    0195          JP      Z,NMIX          ;IF FLG = 'R' -> NMIX
00C6  3D        0196          DEC     A               ;ELSE IF FLG NOT = 'S'
00C7  20EE      0197          JR      NZ,NMI1A        ; TREAT AS GLITCH ON RESET
00C9  F1        0198          POP     AF              ;ELSE MUST BE MONITOR SS
00CA  E1        0199          POP     HL
00CB  1811      0200          JR      FPM1
                0201
```

```
                        0202  *H FRONT PANEL ROUTINES
                        0203
                        0204  ;GETC - GET A CHARACTER FROM KEYBOARD
                        0205  ;WATCH FOR CTRL Z
                        0206
00CD   F702             0207  GETC:   EMT     KBDIN
00CF   28FC             0208          JR      Z,GETC          ;NO CHAR AVAILABLE
00D1   FE1A             0209          CP      CTRLZ
00D3   C0               0210          RET     NZ              ;DROP THRO' ON CTRL Z
                        0211
                        0212  ;HERE BEGINS THE FRONT PANEL EMULATOR
                        0213  ;
                        0214  ;       "AND NOW I SEE WITH EYE SERENE
                        0215  ;        THE VERY PULSE OF THE MACHINE."
                        0216  ;                       WILLIAM WORDSWORTH
                        0217  ;
                        0218  ;SAVE THE CURRENT REGS ON THE MONITOR STACK
                        0219  ;IN THE ORDER:  PC/SP/IY/IX/HL/DE/BC'/AF'
                        0220
00D4   E5               0221          PUSH    HL              ;SAVE AS 'BREAK' DOES
00D5   F5               0222          PUSH    AF
00D6   D5               0223  FPM:    PUSH    DE              ;ENTRY FROM 'BREAK'
00D7   C5               0224          PUSH    BC
00D8   F70D             0225          EMT     GRAFIX          ;CLEAR SCREEN
00DA   C1               0226          POP     BC
00DB   D1               0227          POP     DE
00DC   F1               0228          POP     AF
00DD   E1               0229          POP     HL
                        0230
00DE   E5               0231  FPM1:   PUSH    HL              ;MAKE ROOM FOR SP
00DF   FDE5             0232          PUSH    IY              ;SAVE ALL
00E1   DDE5             0233          PUSH    IX
00E3   E5               0234  FPM2:   PUSH    HL              ;ENTRY FROM 'SWITCH'
00E4   D5               0235          PUSH    DE
00E5   C5               0236          PUSH    BC
00E6   F5               0237          PUSH    AF
00E7   211000           0238          LD      HL,16           ;CALCULATE OLD SP
00EA   39               0239          ADD     HL,SP
00EB   54               0240          LD      D,H             ;DE <- OLD SP
00EC   5D               0241          LD      E,L
00ED   2B               0242          DEC     HL
00EE   2B               0243          DEC     HL
00EF   2B               0244          DEC     HL
00F0   72               0245          LD      (HL),D          ;SAVE OLD SP AT (HL)
00F1   2B               0246          DEC     HL
00F2   73               0247          LD      (HL),E
                        0248
                        0249  ;RDIS - DISPLAY THE REGISTERS
                        0250  ;THIS IS THE RE-ENTRY POINT IN FP MODE: ITS ADDR IS PUSHED
                        0251  ;AT 'CMD:' AND MOST FP ROUTINES RETURN HERE WITH A 'RET'.
                        0252  ;IX STARTS BY POINTING TO THE TOP OF THE SAVED REGS (OLD PC)
                        0253  ;AND ENDS 2 BYTES BELOW SAVED AF. (RPTR) IS THE OFFSET
                        0254  ;(RANGE 2-16) FROM THE LATTER POSITION OF IX TO THE CURRENT
                        0255  ;REG. IY IS A POINTER TO THE REG NAME STRING.
```

```
                        0256
00F3   FD218001         0257  RDIS:   LD      IY,RNSTR        ;IY <- PTR TO REG NAMES
00F7   DD210E00         0258          LD      IX,14           ;IX <- PTR TO TOP OF SAVED REGS
00FB   DD39             0259          ADD     IX,SP
00FD   3E21             0260          LD      A,'!'           ;INDICATE FP MODE
00FF   CD3A05           0261          CALL    AOUT            ;AFTER CR
0102   F70B             0262          EMT     OPNWT           ;###
0104   21160A           0263          LD      HL,0A16H
0107   365E             0264          LD      (HL),UPARR
0109   218A0B           0265          LD      HL,0B8AH
010C   365D             0266          LD      (HL),RARR
010E   21920B           0267          LD      HL,0B92H
0111   365B             0268          LD      (HL),LARR
0113   214008           0269          LD      HL,840H         ;HL <- START OF REG DISPLAY
0116   0608             0270          LD      B,8
0118   3A0940           0271  RD1:    LD      A,(RPTR)        ;TEST WHICH REG IS POINTED TO
011B   CB2F             0272          SRA     A               ;(RPTR) = 2*B
011D   B8               0273          CP      B
011E   3680             0274          LD      (HL),BLANK      ;ASSUME IT'S NOT THIS ONE
0120   2002             0275          JR      NZ,RD2          ;IF IT'S NOT
0122   365D             0276          LD      (HL),RARR       ;ELSE DISPLAY ARROW
0124   23               0277  RD2:    INC     HL              ;LEAVE A SPACE
0125   37               0278          SCF
0126   CD7401           0279          CALL    RNAM            ;OUTPUT REG NAME
0129   CD7401           0280          CALL    RNAM
012C   3E04             0281          LD      A,4             ;MIGHT IT BE THE ALT SET?
012E   B8               0282          CP      B
012F   CCFA04           0283          CALL    Z,CLSOPN        ;###PAUSE IF B = 4
0132   380A             0284          JR      C,RD3           ;IF PC/SP/IY/IX
0134   3A0A0D           0285          LD      A,(RSET)        ;(RSET) = 0 FOR STANDARD SET
0137   B7               0286          OR      A
0138   3680             0287          LD      (HL),BLANK      ;ASSUME NOT
013A   2802             0288          JR      Z,RD3           ;IF STANDARD SET
013C   3627             0289          LD      (HL),''''       ;ELSE APPEND A '
013E   23               0290  RD3:    INC     HL              ;LEAVE A GAP
013F   23               0291          INC     HL
0140   DD5601           0292          LD      D,(IX+1)        ;DE <- REG CONTENTS
0143   DD5E00           0293          LD      E,(IX+0)
0146   DD2B             0294          DEC     IX
0148   DD2B             0295          DEC     IX
014A   CD8F06           0296          CALL    .DEOUT          ;DISPLAY CONTENTS
014D   100C             0297          DJNZ    RD5             ;FOR ALL BUT AF, ALSO DISPLAY (REG)
                        0298
014F   23               0299          INC     HL              ;AF, DISPLAY FLAGS
0150   7B               0300          LD      A,E             ;A <- FLAG REG
0151   0608             0301          LD      B,8
0153   17               0302  RD4:    RLA                     ;CY <- NEXT FLAG
0154   CD7401           0303          CALL    RNAM            ;DISPLAY FLAG IF CY SET
0157   10FA             0304          DJNZ    RD4
0159   183D             0305          JR      MDIS
                        0306
015B   E5               0307  RD5:    PUSH    HL              ;SAVE VT ADDR
015C   21FCFF           0308          LD      HL,-4           ;DISPLAY (REG-4) TO (REG+3)
015F   19               0309          ADD     HL,DE
```

```
0160  EB        0310          EX    DE,HL          ;(DE) WILL BE DISPLAYED
0161  E1        0311          POP   HL
0162  48        0312          LD    C,B            ;SAVE B
0163  0608      0313          LD    B,8
0165  23        0314  RD6:    INC   HL
0166  1A        0315          LD    A,(DE)         ;OUTPUT (REG)
0167  13        0316          INC   DE
0168  CD9406    0317          CALL  .BYTEO
016B  10F8      0318          DJNZ  RD6
016D  41        0319          LD    B,C            ;RESTORE B
016E  111F00    0320          LD    DE,40H-33      ;DE <- DISTANCE TO NEXT LINE ON VT
0171  19        0321          ADD   HL,DE          ;UPDATE VT ADDR
0172  18A4      0322          JR    RD1
              0323
0174  3680      0324  RNAM:   LD    (HL),BLANK
0176  3004      0325          JR    NC,RN1         ;SKIP IF NO CARRY
0178  FD4E00    0326          LD    C,(IY+0)       ;ELSE OUTPUT NEXT CHAR FROM RNSTR
017B  71        0327          LD    (HL),C
017C  23        0328  RN1:    INC   HL
017D  FD23      0329          INC   IY
017F  C9        0330          RET
              0331
0180  50435350  0332  RNSTR:  DEFM  'PCSPIYIXHLDEBCAFSZ H VNC'
              0333
              0334  ;MDIS - DISPLAY MEMORY AROUND MPTR
              0335
0198  CDFA04    0336  MDIS:   CALL  CLSOPN         ;###PAUSE TO AVOID BLINK
019B  2A0740    0337          LD    HL,(MPTR)      ;DE <- (MPTR)-12
019E  11F4FF    0338          LD    DE,-12
01A1  19        0339          ADD   HL,DE
01A2  EB        0340          EX    DE,HL
01A3  21800A    0341          LD    HL,0A80H       ;HL <- START OF DISPLAY
01A6  0604      0342          LD    B,4            ;4 COLUMNS
01A8  48        0343  MD1:    LD    C,B            ;COUNTED BY C
01A9  79        0344          LD    A,C            ;###IF C = 2
01AA  FE02      0345          CP    2              ;###
01AC  CCFA04    0346          CALL  Z,CLSOPN       ;###PAUSE TO AVOID BLINK
01AF  0608      0347          LD    B,8            ;8 ROWS
01B1  CD8F06    0348  MD2:    CALL  .DEOUT         ;OUTPUT ADDRESS
01B4  23        0349          INC   HL             ;LEAVE A BLANK
01B5  1A        0350          LD    A,(DE)         ;GET CONTENT
01B6  CD9406    0351          CALL  .BYTEO         ;OUTPUT IT
01B9  13        0352          INC   DE             ;POINT TO NEXT
01BA  D5        0353          PUSH  DE
01BB  113900    0354          LD    DE,57          ;BUMP VT POINTER
01BE  19        0355          ADD   HL,DE
01BF  D1        0356          POP   DE
01C0  10EF      0357          DJNZ  MD2            ;REPEAT FOR 8 ROWS
01C2  D5        0358          PUSH  DE
01C3  110BFE    0359          LD    DE,-200H+11    ;START OF NEXT COLUMN
01C6  19        0360          ADD   HL,DE
01C7  D1        0361          POP   DE
01C8  41        0362          LD    B,C
01C9  10DD      0363          DJNZ  MD1            ;REPEAT FOR 4 COLUMNS
```

```
01CB   F70C      0364          EMT     CLOSE           ;###
                 0365
                 0366  ;CMD - GET MODIFIED VALUE FOR MPTR OR REG, PARSE COMMAND
                 0367
01CD   21F300    0368  CMD:    LD      HL,RDIS         ;RETURN ADDRESS
01D0   E5        0369          PUSH    HL
01D1   F713      0370          EMT     GETHEX
01D3   3E2E      0371          LD      A,'.'           ;IF TERMINATOR WAS '.'
01D5   90        0372          SUB     B               ;THIS IS A REG OPERATION
01D6   2830      0373          JR      Z,SETR
01D8   79        0374          LD      A,C             ;GET NUMBER OF HEX CHARS
01D9   B7        0375          OR      A
01DA   2809      0376          JR      Z,CMD1          ;IF THERE WERE NONE
01DC   7D        0377          LD      A,L             ;A <- BYTE
01DD   2A0740    0378          LD      HL,(MPTR)
01E0   F70B      0379          EMT     DFNWT           ;###IN CASE (MPTR) IS VT
01E2   77        0380          LD      (HL),A          ;STORE IT
01E3   F70C      0381          EMT     CLOSE           ;###
01E5   78        0382  CMD1:   LD      A,B             ;A <- TERMINATOR
01E6   213002    0383          LD      HL,TBL1
01E9   1600      0384          LD      D,0
01EB   42        0385          LD      B,D
01EC   0E08      0386          LD      C,8             ;BC <- TBL LENGTH (1ST)
01EE   EF37      0387          CALR    LOOKUP
01F0   200C      0388          JR      NZ,CMD2         ;IF NO MATCH
01F2   2A0740    0389          LD      HL,(MPTR)       ;UPDATE MPTR
01F5   01F0FF    0390          LD      BC,-16          ;VALUES FROM TBL ARE OFFSET
01F8   09        0391          ADD     HL,BC           ; TO SAVE INSTRUCTIONS
01F9   19        0392          ADD     HL,DE           ;ADD UPDATE
01FA   220740    0393  SVM:    LD      (MPTR),HL
01FD   C9        0394          RET
                 0395
01FE   0E18      0396  CMD2:   LD      C,TBL2LN        ;BC <- TBL LENGTH (2ND)
0200   EF25      0397          CALR    LOOKUP
0202   E0        0398          RET     PO              ;NOTHING FOUND
0203   215002    0399          LD      HL,SETM         ;CALCULATE REL ADDR
0206   19        0400          ADD     HL,DE
0207   E9        0401          JP      (HL)
                 0402
                 0403  ;SETR - MODIFY REG OR MOVE REG POINTER
                 0404
0208   57        0405  SETR:   LD      D,A             ;CLEAR D FOR LATER (A = 0)
0209   B9        0406          CP      C               ;TEST IF ANY HEX
020A   3A0940    0407          LD      A,(RPTR)        ;A <- OFFSET FROM IX TO REG
020D   280A      0408          JR      Z,SR1           ;IF NO HEX
020F   5F        0409          LD      E,A             ;ELSE LOAD REG
0210   DD19      0410          ADD     IX,DE           ;POINT TO REG
0212   DD7401    0411          LD      (IX+1),H        ;REG <- HL
0215   DD7500    0412          LD      (IX+0),L
0218   C9        0413          RET                     ;OR 'NOP'
0219   3D        0414  SR1:    DEC     A               ;DECR OFFSET, TO NEXT REG
021A   3D        0415          DEC     A
021B   2002      0416          JR      NZ,SR2          ;UNLESS OFF THE BOTTOM
021D   3E10      0417          LD      A,16            ;POINT TO PC
```

```
021F  FE0E      0418  SR2:    CP      14              ;SKIP SP
0221  28F6      0419          JR      Z,SR1
0223  320940    0420          LD      (RPTR),A        ;SAVE OFFSET
0226  C9        0421          RET
                0422
                0423  ;LOOKUP - SEARCH TABLE POINTED TO BY HL FOR BC BYTES,
                0424  ;RETURNING ENTRY IN E
                0425  ;Z SET FOR A MATCH, P ODD IF OFF THE END
                0426
0227  EDA1      0427  LOOKUP: CPI
0229  5E        0428          LD      E,(HL)
022A  C8        0429          RET     Z               ;IF MATCH
022B  EDA1      0430          CPI
022D  E0        0431          RET     PO              ;IF OFF THE END
022E  18F7      0432          JR      LOOKUP
                0433
                0434  ;TBL - FRONT PANEL SWITCHES
                0435
0230  0D        0436  TBL1:   DEFB    CR              ;FORWARD ONE
0231  11        0437          DEFB    17
0232  2D        0438          DEFB    '-'             ;BACK ONE
0233  0F        0439          DEFB    15
0234  0A        0440          DEFB    LF              ;FORWARD EIGHT
0235  18        0441          DEFB    24
0236  2F        0442          DEFB    '/'             ;BACK EIGHT
0237  08        0443          DEFB    8
0238  47        0444  TBL2:   DEFB    'G'             ;GET AND FIND PATTERN
0239  87        0445          DEFB    GETPAT-SETM
023A  48        0446          DEFB    'H'             ;HEX CALCULATOR
023B  6B        0447          DEFB    HEXCAL-SETM
023C  49        0448          DEFB    'I'             ;MPTR <- CURRENT WORD
023D  05        0449          DEFB    IND-SETM
023E  4A        0450          DEFB    'J'             ;CLEAR STACK, JUMP
023F  C3        0451          DEFB    FPJMP-SETM
0240  4B        0452          DEFB    'K'             ;CONTINUE EXECUTION
0241  39        0453          DEFB    CONTIN-SETM
0242  4D        0454          DEFB    'M'             ;SET MPTR
0243  00        0455          DEFB    SETM-SETM
0244  4E        0456          DEFB    'N'             ;FIND NEXT PATTERN
0245  9E        0457          DEFB    GPNXT-SETM
0246  50        0458          DEFB    'P'             ;FILL AND TEST MEMORY
0247  47        0459          DEFB    PUT-SETM
0248  52        0460          DEFB    'R'             ;MPTR <- MPTR + CURRENT OFFSET
0249  0E        0461          DEFB    REL-SETM
024A  53        0462          DEFB    'S'             ;MOVE MEMORY
024B  E5        0463          DEFB    MOVER-SETM
024C  57        0464          DEFB    'W'             ;SWITCH REGISTER SET
024D  20        0465          DEFB    SWITCH-SETM
024E  5A        0466          DEFB    'Z'             ;SINGLE STEP
024F  31        0467          DEFB    STEP-SETM
      0018      0468  TBL2LN  EQU     $-TBL2
                0469
                0470  ;SETM - PROMPT AND GET NEW VALUE FOR MPTR
                0471
```

```
0250  CD0006    0472  SETM:   CALL    PGTHEX
0253  18A5      0473          JR      SVM
                0474
                0475  ;IND - SET MPTR TO CURRENT WORD ADDRESS
                0476
0255  2A0740    0477  IND:    LD      HL,(MPTR)
0258  5E        0478          LD      E,(HL)
0259  23        0479          INC     HL
025A  56        0480          LD      D,(HL)
025B  EB        0481          EX      DE,HL
025C  189C      0482          JR      SVM
                0483
                0484  ;REL - SET MPTR TO CURRENT RELATIVE ADDRESS
                0485
025E  2A0740    0486  REL:    LD      HL,(MPTR)
0261  EF02      0487          CALR    RTOA            ;HL <- HL + (HL) + 1
0263  1895      0488          JR      SVM
                0489
                0490  ;RTOA - LOAD HL WITH HL + (HL) + 1 (REL TO ABS)
                0491  ;DESTROYS AF AND DE
                0492
0265  5E        0493  RTOA:   LD      E,(HL)          ;E <- OFFSET
0266  1600      0494          LD      D,0             ;EXTEND SIGN TO D
0268  CB7B      0495          BIT     7,E
026A  2801      0496          JR      Z,RA1
026C  15        0497          DEC     D
026D  19        0498  RA1:    ADD     HL,DE
026E  23        0499          INC     HL
026F  C9        0500          RET
                0501
                0502  ;SWITCH - SWITCH REGISTER SET BEING DISPLAYED
                0503
0270  3A0A40    0504  SWITCH: LD      A,(RSET)        ;TOGGLE THE SWITCH
0273  2F        0505          CPL
0274  320A40    0506          LD      (RSET),A
0277  F1        0507          POP     AF              ;POP RET ADDR
0278  F1        0508          POP     AF              ;RESTORE CURRENT SET
0279  C1        0509          POP     BC
027A  D1        0510          POP     DE
027B  E1        0511          POP     HL
027C  08        0512          EX      AF,AF'          ;EXCHANGE REGISTERS
027D  D9        0513          EXX
027E  C3E300    0514          JP      FPM2            ;RE-ENTER TO DISPLAY OTHER SET
                0515
                0516  ;STEP - ENABLE SINGLE INSTRUCTION MODE
                0517  ;       MODE IS CLEARED ON NMI
                0518
0281  210340    0519  STEP:   LD      HL,MASK         ;SET SS BIT IN MASK
0284  CB8E      0520          RES     SINGLE,(HL)
0286  23        0521          INC     HL              ;POINT TO SS FLG
0287  3653      0522          LD      (HL),'S'        ;SET MONITOR SS FLAG
                0523
                0524  ;CONTIN - CONTINUE EXECUTION AT SAVED PC
                0525  ;         CALL UPDATE TO ENABLE
```

```
                        0526  ;          SINGLE STEP LOGIC IF SET IN 'MASK'
                        0527
0289    F1              0528  CONTIN: POP     AF              ;POP FP RET ADDR
028A    F1              0529          POP     AF              ;POP CURRENT REGS
028B    C1              0530          POP     BC
028C    D1              0531          POP     DE
028D    E1              0532          POP     HL
028E    DDE1            0533          POP     IX
0290    F712            0534          EMT     UPDATE          ;SS ON IF SET
0292    FDE1            0535          POP     IY
0294    33              0536          INC     SP
0295    33              0537          INC     SP
0296    C9              0538          RET
                        0539
```

```
                    0540  *E
                    0541  ;PUT - FILLS AND TESTS MEMORY BETWEEN LIMITS
                    0542
0297  CDE203        0543  PUT:    CALL    FSTLST          ;DE <- FIRST, HL <- DIFF
029A  23            0544          INC     HL              ;LENGTH INCLUSIVE
029B  E5            0545          PUSH    HL              ;SAVE COUNT
029C  2134C0        0546          LD      HL,MSGP
029F  CDF805        0547          CALL    MGTHEX          ;GET FILL BYTE
02A2  7D            0548          LD      A,L             ;A = FILLER
02A3  EB            0549          EX      DE,HL           ;HL = START
02A4  C1            0550          POP     BC              ;BC = NO. OF ADDRS TO FILL
                    0551
02A5  77            0552  PT1:    LD      (HL),A          ;FILL MEMORY
02A6  EDA1          0553          CPI                     ;THEN CHECK IT
02A8  280E          0554          JR      Z,PT2           ;IF (HL) MATCHES A
02AA  F5            0555          PUSH    AF              ;ELSE ERR, SAVE PARITY FLAG
02AB  E5            0556          PUSH    HL              ;SAVE NEXT ADDR TO TEST
02AC  2B            0557          DEC     HL              ;(HL) IS NOW INCORRECT BYTE
02AD  220740        0558          LD      (MPTR),HL       ;SO IT'S SEEN ON FP
02B0  CDC905        0559          CALL    ERROUT          ;ERROR MSG
02B3  E1            0560          POP     HL              ;RESTORE NEXT ADDR
02B4  F1            0561          POP     AF              ;RESTORE PARITY FLAG
02B5  CDDE00        0562          CALL    FPM1            ;FP ON, CONTINUE TEST AFTER 'K'
02B8  E0            0563  PT2:    RET     PO              ;UNTIL BC = 0
02B9  18EA          0564          JR      PT1
```

```
                        0565  *E
                        0566  ;HEXCAL - HEXADECIMAL CALCULATOR
                        0567
02BB   CDFD05           0568  HEXCAL: CALL    CGTHEX          ;GET X
02BE   E5               0569          PUSH    HL              ;(SP) <- X
02BF   EB               0570          EX      DE,HL           ;DE <- X
02C0   CDFD05           0571          CALL    CGTHEX          ;GET Y
02C3   E5               0572          PUSH    HL              ;(SP) <- Y
02C4   19               0573          ADD     HL,DE           ;HL <- X+Y
02C5   EB               0574          EX      DE,HL           ;DE <- X+Y
02C6   21C60D           0575          LD      HL,0DC6H        ;HL <- VT ADDR
02C9   CD8806           0576          CALL    DEOUTW          ;DISPLAY SUM
02CC   D1               0577          POP     DE              ;GET Y
02CD   E3               0578          EX      (SP),HL         ;GET X, SAVE VT PTR
02CE   B7               0579          OR      A               ;CLEAR CY
02CF   ED52             0580          SBC     HL,DE           ;HL <- X-Y
02D1   EB               0581          EX      DE,HL           ;DE <- X-Y
02D2   E1               0582          POP     HL              ;RESTORE VT PTR
02D3   23               0583          INC     HL
02D4   C38806           0584          JP      DEOUTW          ;DISP DIFF, RET TO FP
                        0585
```

```
                        0586  *E
                        0587  ;GETPAT - ENTRY TO GET PATN FROM KBD, THEN SEARCH
                        0588
02D7   114C40           0589  GETPAT: LD      DE,PATN         ;DE <- PATN ADDR
02DA   AF               0590          XOR     A
02DB   324B40           0591  GP1:    LD      (PATL),A        ;UPDATE PATN LENGTH
02DE   CDFD05           0592          CALL    CGTHEX          ;POSSIBLY GET A BYTE
02E1   79               0593          LD      A,C             ;TEST FOR ONE
02E2   B7               0594          OR      A
02E3   2809             0595          JR      Z,GPNXT         ;JUMP IF NO BYTE
02E5   7D               0596          LD      A,L             ;ELSE STORE IT
02E6   12               0597          LD      (DE),A
02E7   13               0598          INC     DE
02E8   3A4B40           0599          LD      A,(PATL)        ;INC LENGTH
02EB   3C               0600          INC     A
02EC   18ED             0601          JR      GP1
                        0602
                        0603  ;GPNXT - GET NEXT OCCURRENCE OF PATN
                        0604
02EE                    0605  GPNXT:
02EE   2A0740           0606  GP2:    LD      HL,(MPTR)       ;START FROM (MPTR)+1
02F1   23               0607          INC     HL
02F2   114C40           0608          LD      DE,PATN         ;DE <- START OF PATTERN
02F5   1A               0609          LD      A,(DE)          ;A <- FIRST BYTE TO FIND
02F6   010000           0610          LD      BC,0            ;SET BC FOR 65K
02F9   EDB1             0611          CPIR                    ;FIND FIRST MATCH
02FB   2B               0612          DEC     HL              ;POINT TO START AND
02FC   220740           0613          LD      (MPTR),HL       ;SAVE MEMORY PTR
02FF   23               0614          INC     HL              ;THEN POINT TO NEXT AGAIN
0300   3A4B40           0615          LD      A,(PATL)        ;A <- PATTERN LENGTH
0303   B7               0616          OR      A               ;/IGNORE ZERO LENGTH
0304   C8               0617          RET     Z               ;/
0305   3D               0618          DEC     A               ;HAVE FOUND FIRST
0306   4F               0619          LD      C,A             ;BC <- LENGTH
0307   AF               0620  GP3:    XOR     A
0308   47               0621          LD      B,A
0309   B1               0622          OR      C               ;WHILE (BC > 0)
030A   C8               0623          RET     Z
030B   13               0624          INC     DE              ;CHECK REST OF PATN
030C   1A               0625          LD      A,(DE)          ;A <- NEXT IN PATN
030D   EDA1             0626          CPI
030F   20DD             0627          JR      NZ,GP2          ;IF NO MATCH, BREAK TO CONTINUE SEARCH
0311   18F4             0628          JR      GP3
                        0629
```

```
                        0630  *H FP/MONITOR ROUTINES
                        0631
                        0632  ;JUMP - PROMPT FOR ADDRESS, THEN
                        0633  ;RESTORE SCROLLER AND GO TO ADDRESS
                        0634
0313   310041           0635  FPJMP:  LD      SP,STACK        ;CLEAN UP STACK
                        0636
0316   CD0006           0637  JUMP:   CALL    PGTHEX          ;GET ADDR
0319   E5               0638          PUSH    HL              ;SAVE FOR RET
                        0639
                        0640  ;SCROLL - RESTORES FULL SCREEN SCROLL
                        0641
031A   010300           0642  .SCROL: LD      BC,3            ;RESTORE SCROLLER
                        0643
                        0644  ;INIT - MOVES BC WORDS FROM INITBK TO RAM SCRATCH AREA
                        0645
031D   21DE07           0646  INIT:   LD      HL,INITBK
0320   110040           0647  IT1:    LD      DE,RAM
0323   EDB0             0648          LDIR
0325   C9               0649          RET
                        0650
                        0651  ;GRAFIX - CLEARS GRAPHICS ('FRONT PANEL') AREA
                        0652  ;AND SETS SCROLLER TO BOTTOM FOUR LINES
                        0653
0326   3E14             0654  .GRAFX: LD      A,20            ;NO. OF LINES
0328   210008           0655          LD      HL,800H         ;TOP OF SCREEN
032B   F70F             0656          EMT     CLEAR
                        0657
                        0658  ;GRAFNC - SETS SCROLLER WITHOUT CLEARING
                        0659
032D   010300           0660  GRAFNC: LD      BC,3
0330   21FD07           0661          LD      HL,GRAFBK
0333   18EB             0662          JR      IT1
                        0663
                        0664  ;MOVER - SEE MOVE
                        0665
0335   186E             0666  MOVER:  JR      MOVE
                        0667
```

```
                        0668  *H MONITOR ROUTINES
                        0669
                        0670  ;       'THE WHITE RABBIT PUT ON HIS SPECTACLES. 'WHERE
                        0671  ;        SHALL I BEGIN, PLEASE YOUR MAJESTY?' HE ASKED.
                        0672  ;        'BEGIN AT THE BEGINNING,' THE KING SAID GRAVELY,
                        0673  ;        'AND GO ON TILL YOU COME TO THE END: THEN STOP.'"
                        0674  ;               LEWIS CARROLL  ALICE'S ADVENTURES IN WONDERLAND
                        0675
                        0676  ;BEGIN - SET STACK AND SYSTEM POINTERS, DROP
                        0677  ;THROUGH TO MAIN MONITOR COMMANDS
                        0678
0337  310041            0679  BEGIN:  LD      SP,STACK
033A  F3                0680          DI
033B  011F00            0681          LD      BC,INITLN       ;COPY INITBK TO RAM
033E  EFDB              0682          CALR    INIT
0340  3EFF              0683          LD      A,0FFH          ;SET REMAINING VECTORS TO OFFH
0342  0621              0684          LD      B,FFLEN
0344  211F40            0685          LD      HL,OUT1
0347  77                0686  BG1:    LD      (HL),A
0348  23                0687          INC     HL
0349  10FC              0688          DJNZ    BG1
034B  36C9              0689          LD      (HL),0C9H       ;SET MONX TO 'RET'
                        0690
034D  3E93              0691          LD      A,MKINIT!1      ;/CLEAR KBD (RESETS PORT0)
034F  32FC0F            0692          LD      (PORT0),A
0352  F712              0693          EMT     UPDATE
0354  3E0C              0694          LD      A,FORM          ;CLEAR SCREEN
0356  F716              0695          EMT     OUTC
                        0696
0358  224340            0697  BG2:    LD      (HIMEM),HL      ;FIND TOP OF RAM
035B  23                0698          INC     HL
035C  7E                0699          LD      A,(HL)          ;SAVE CONTENT
035D  2F                0700          CPL
035E  77                0701          LD      (HL),A          ;TEST IF COMPL READS BACK
035F  BE                0702          CP      (HL)
0360  2F                0703          CPL
0361  77                0704          LD      (HL),A          ;RESTORE ORIGINAL CONTENT
0362  28F4              0705          JR      Z,BG2           ;CONTINUE UNTIL END OF RAM
                        0706
0364  3A0010            0707          LD      A,(NXTROM)      ;TEST FOR ADDITIONAL ROM
0367  B7                0708          OR      A               ;IF IT CONTAINS ZERO
0368  CC0010            0709          CALL    Z,NXTROM        ; CALL IT
                        0710
036B  21D607            0711          LD      HL,MSGV         ;OUTPUT VERSION
036E  CDCC05            0712          CALL    MOUT
                        0713
                        0714  ;CONTC - DO PARTIAL RESET, THEN DROP THROUGH TO MONITOR
                        0715
0371  310041            0716  CONTC:  LD      SP,STACK
0374  010400            0717          LD      BC,CCLEN        ;RESET SCROLLER TO FULL SCREEN
0377  EFA4              0718          CALR    INIT            ;AND INITIALISE PORT0 MASK
                        0719
                        0720  ;ZMON - MONITOR COMMAND LOOP
                        0721
```

```
0379  3E5D      0722  ZMON:   LD      A,RARR          ;ISSUE CR, MONITOR PROMPT
037B  CD3A05    0723          CALL    AOUT
037E  CD4040    0724          CALL    MONX            ;CHECK FOR EXTENDED MONITOR
0381  CDCD00    0725          CALL    GETC
0384  F716      0726          EMT     OUTC
0386  FE4C      0727          CP      'L'             ;LOAD MEMORY FROM CASSETTE
0388  283F      0728          JR      Z,LOAD
038A  FE44      0729          CP      'D'             ;DUMP MEMORY TO CASSETTE
038C  CA6204    0730          JP      Z,SAVE          ;/POSSIBLY JR
038F  FE4A      0731          CP      'J'             ;GO TO ADDRESS
0391  2883      0732          JR      Z,JUMP
0393  FE43      0733          CP      'C'             ;REENTER USER PROG
0395  2807      0734          JR      Z,RNTR
0397  FE53      0735          CP      'S'             ;MOVE MEMORY
0399  CCA503    0736          CALL    Z,MOVE          ; MUST BE LAST IN LIST
039C  18DB      0737          JR      ZMON            ; BECAUSE MOVE DOES A 'RET'
                0738
                0739  ;RNTR - REENTER USER PROGRAM AT (ADDR) + 3
                0740
039E  2A0B40    0741  RNTR:   LD      HL,(ADDR)
03A1  23        0742          INC     HL
03A2  23        0743          INC     HL
03A3  23        0744          INC     HL
03A4  E9        0745          JP      (HL)
                0746
                0747  ;MOVE - MOVE BLOCK OF MEMORY UP OR DOWN
                0748  ;       PROMPTS FOR START (OLD S) AND FINISH (OLD F),
                0749  ;        THEN NEW START (NEW S)
                0750
03A5  CDE205    0751  MOVE:   CALL    FSTLST          ;DE <- OLD S, HL <- (OLD F-OLD S)
03A8  E5        0752          PUSH    HL              ;(SP) <- LEN-1
03A9  210600    0753          LD      HL,MSGM         ;GET NEW S
03AC  CDF805    0754          CALL    MGTHEX          ;HL <- NEW S
03AF  EB        0755          EX      DE,HL           ;HL <- OS, DE <- NS
03B0  C1        0756          POP     BC              ;BC <- LEN-1
03B1  E5        0757          PUSH    HL              ;(SP) <- OLD S
03B2  B7        0758          OR      A               ;CLEAR CY
03B3  ED52      0759          SBC     HL,DE           ;CY->UP, NO CY->DOWN
03B5  E1        0760          POP     HL              ;HL <- OLD S
03B6  F70B      0761          EMT     OPNWT           ;### ALLOW VT MOVES
03B8  3009      0762          JR      NC,MV1          ;IF MOVE DOWN
03BA  09        0763          ADD     HL,BC           ;HL <- OLD F
03BB  EB        0764          EX      DE,HL
03BC  09        0765          ADD     HL,BC
03BD  EB        0766          EX      DE,HL           ;DE <- NEW F
03BE  03        0767          INC     BC              ;MAKE LENGTH INCLUSIVE
03BF  EDB8      0768          LDDR                    ;MOVE UP FROM TOP
03C1  1803      0769          JR      MV2
03C3  03        0770  MV1:    INC     BC              ;MAKE LENGTH INCLUSIVE
03C4  EDB0      0771          LDIR                    ;MOVE DOWN FROM BOTTOM
03C6  F70C      0772  MV2:    EMT     CLOSE           ;###
03C8  C9        0773          RET
                0774
```

```
                        0775  *H ABSOLUTE LOADER & DUMP
                        0776
                        0777  ;LOAD - LOADS TAPE IN RML ABSOLUTE FORMAT
                        0778
03C9    CD5406          0779  LOAD:   CALL    GETNAM          ;GET NAME TO FIND
03CC    CD4505          0780          CALL    CROUT
                        0781                                  ;START TAPE
03CF    210000          0782  FIND:   LD      HL,0            ;/SET RECORD ADDR TO ZERO
03D2    CD4C04          0783          CALL    RECNO           ;/
03D5    CD2904          0784          CALL    GETHDR
03D8    3C              0785          INC     A               ;SET Z IF NAME REC (TYPE -1)
03D9    20F4            0786          JR      NZ,FIND         ;ELSE CONTINUE SEARCH
                        0787
03DB    214C40          0788          LD      HL,PATN         ;HL <- BUFFER ADDR FOR NAME
03DE    E5              0789          PUSH    HL              ;SAVE THE ADDR
03DF    CD3B04          0790          CALL    GET             ;READ THE NAME REC
03E2    2025            0791          JR      NZ,LDERR        ;CKSUM ERROR
03E4    36FF            0792          LD      (HL),-1         ;PUT EOS TO BUF
03E6    E1              0793          POP     HL              ;HL <- START OF NAME
03E7    CDCF05          0794          CALL    .MSG            ;DISPLAY IT (HL UNCHANGED)
03EA    CD4505          0795          CALL    CROUT
03ED    114540          0796          LD      DE,NAME         ;DE <- START OF WANTED NAME
03F0    0606            0797          LD      B,6             ;COMPARE 6 CHARS
03F2    1A              0798  COMP:   LD      A,(DE)          ;GET CHAR FROM TARGET
03F3    BE              0799          CP      (HL)            ;COMPARE CHAR FROM THIS NAME
03F4    20D9            0800          JR      NZ,FIND         ;JUMP IF NO MATCH
03F6    13              0801          INC     DE              ;INC POINTERS
03F7    23              0802          INC     HL
03F8    10F8            0803          DJNZ    COMP            ;REPEAT IF MORE
                        0804
03FA    CD2904          0805  GETREC: CALL    GETHDR
03FD    E5              0806          PUSH    HL              ;SAVE ADDR
03FE    B7              0807          OR      A               ;TEST TYPE
03FF    2017            0808          JR      NZ,GR1          ;IF NOT TYPE 0
0401    CD3B04          0809          CALL    GET             ;LOAD DATA
0404    E1              0810          POP     HL              ;/RESTORE ADDR
0405    EF45            0811          CALR    RECNO           ;/AND DISPLAY IT
0407    28F1            0812          JR      Z,GETREC        ;Z MEANS CKSUM OK
                        0813
0409    CDC905          0814  LDERR:  CALL    ERROUT          ;OUTPUT ERROR MSG
040C    CDCD00          0815  LDER1:  CALL    GETC            ;WAIT FOR KBD
040F    FE4C            0816          CP      'L'
0411    20F9            0817          JR      NZ,LDER1        ;WAIT FOR 'L'
0413    CD4505          0818          CALL    CROUT
0416    18E2            0819          JR      GETREC          ;CONTINUE LOADING
                        0820
0418    3D              0821  GR1:    DEC     A               ;TEST TYPE
0419    200B            0822          JR      NZ,GR2          ;IF NOT TYPE 1
041B    EF28            0823          CALR    TI              ;EOF, GET CKSUM
041D    E1              0824          POP     HL              ;/RESTORE ADDR
041E    EF2C            0825          CALR    RECNO           ;/AND DISPLAY IT
0420    20E7            0826          JR      NZ,LDERR
                        0827                                  ;STOP TAPE
0422    220F00          0828          LD      (ADDR),HL       ;SAVE START ADDR
```

```
0425  E9        0829          JP      (HL)            ;GO TO ADDRESS
                0830
0426  E1        0831  GR2:    POP     HL              ;/SKIP REC IF NOT TYPES 0 OR 1
0427  18D1      0832          JR      GETREC          ;CONTINUE LOADING
                0833
                0834  ;GETHDR - GET RECORD HEADER
                0835  ; RETURN WITH HL=ADDRESS, B=LENGTH AND A=TYPE
                0836
0429  3E4D      0837  GETHDR: LD      A,SOH           ;WAIT FOR START OF RECORD
042B  F711      0838          EMT     GETSYN
042D  AF        0839          XOR     A               ;CLEAR CKSUM
042E  4F        0840          LD      C,A
042F  EF14      0841          CALR    TI              ;GET LENGTH
0431  47        0842          LD      B,A
0432  EF11      0843          CALR    TI              ;GET ADDR H
0434  67        0844          LD      H,A
0435  EF0E      0845          CALR    TI              ;GET ADDR L
0437  6F        0846          LD      L,A
0438  EF0B      0847          CALR    TI              ;GET RECORD TYPE
043A  C9        0848          RET
                0849
                0850  ;GET - READ B BYTES FROM TAPE TO (HL), RETURNING
                0851  ;       WITH Z SET FROM CKSUM
                0852
043B  EF08      0853  GET:    CALR    TI
043D  77        0854          LD      (HL),A
043E  BE        0855          CP      (HL)
043F  2801      0856          JR      Z,GT1           ;IF BYTE LOADED
0441  FF        0857          RST     38H             ;ELSE BREAK
0442  23        0858  GT1:    INC     HL
0443  10F6      0859          DJNZ    GET             ;DROP THRO' TO TI FOR CKSUM
                0860
                0861  ;TI - READ BYTE FROM TAPE INTO A, FORM CKSUM
                0862
0445  F704      0863  TI:     EMT     GETBYT
0447  57        0864          LD      D,A             ;D <- BYTE
0448  81        0865          ADD     A,C             ;ADD CKSUM TO BYTE
0449  4F        0866          LD      C,A             ;RESTORE CKSUM
044A  7A        0867          LD      A,D             ;RESTORE BYTE
044B  C9        0868          RET                     ;WITH Z FLAG FROM CKSUM
                0869
                0870  ;RECNO - INDICATE TAPE ACTIVITY, DISPLAY RECORD ADDRESS
                0871
044C  E5        0872  RECNO:  PUSH    HL              ;STANDARD SAVE SEQUENCE
044D  D5        0873          PUSH    DE
044E  C5        0874          PUSH    BC
044F  F5        0875          PUSH    AF
0450  EB        0876          EX      DE,HL           ;/DE <- ADDR (HL ON ENTRY)
0451  21C00B    0877          LD      HL,0DC0H        ;HL <- CURSOR ADDR
0454  F70B      0878          EMT     OPNWT           ;###
0456  7E        0879          LD      A,(HL)          ;BLINK CURSOR
0457  EE7F      0880          XOR     CURSOR
0459  77        0881          LD      (HL),A
045A  23        0882          INC     HL              ;/LEAVE A SPACE
```

```
045B   23          0883          INC     HL              ;/
045C   CD8A06      0884          CALL    DEOUTC          ;/### AND CLOSE
045F   C3C405      0885          JP      RET1            ;STANDARD EXIT
                   0886
```

```
                    0887  *E
                    0888  ;SAVE - DUMPS MEMORY TO TAPE WITH OPTIONAL START ADDR
                    0889
0462  CD5406        0890  SAVE:   CALL    GETNAM          ;GET A NAME
0465  CDE205        0891          CALL    FSTLST          ;DE <- FIRST ADDR, BC <- LAST
0468  C5            0892          PUSH    BC              ;(SP) <- LAST
0469  212300        0893          LD      HL,MSGS         ;GET AUTOSTART ADDR
046C  CDF805        0894          CALL    MGTHEX          ;HL <- START
046F  E3            0895          EX      (SP),HL         ;(SP) <- START ADDR
0470  E5            0896          PUSH    HL              ;SAVE LAST ADDR
0471  D5            0897          PUSH    DE              ;SAVE FIRST ADDR
0472  CD4505        0898          CALL    CROUT
0475  210000        0899          LD      HL,0            ;/CLEAR REC ADDR
0478  EFD2          0900          CALR    RECNO           ;/INDICATE ACTIVITY
                    0901                                  ;START TAPE
047A  CD9607        0902          CALL    SEC5            ;5 SEC DELAY
                    0903
047D  214540        0904          LD      HL,NAME         ;OUTPUT NAME
0480  16FF          0905          LD      D,-1            ;REC TYPE -1 (NAME)
0482  0606          0906          LD      B,6
0484  CDB404        0907          CALL    PUTRNA          ;PUTREC, NO ADDR
0487  D1            0908          POP     DE              ;RESTORE DUMP LIMITS
0488  E1            0909          POP     HL
                    0910
0489  E5            0911  SV1:    PUSH    HL              ;(SP) <- LAST ADDR
048A  067A          0912          LD      B,RECLEN-6      ;B <- RECORD SIZE
048C  AF            0913          XOR     A               ;CLEAR CY
048D  ED52          0914          SBC     HL,DE           ;GET LENGTH TO DUMP
048F  3813          0915          JR      C,SV3           ;DUMP COMPLETE (BECAUSE INCL)
0491  BC            0916          CP      H               ;IF LENGTH > 255
0492  2006          0917          JR      NZ,SV2          ; CONTINUE
0494  7D            0918          LD      A,L             ;IF LENGTH >= 1 REC
0495  B8            0919          CP      B
0496  3002          0920          JR      NC,SV2          ; CONTINUE
0498  47            0921          LD      B,A             ;ELSE DUMP REMAINDER
0499  04            0922          INC     B               ; INCLUSIVELY
049A  EB            0923  SV2:    EX      DE,HL           ;NOW HL <- CURRENT ADDR
049B  1600          0924          LD      D,0             ;REC TYPE 0 (DATA)
049D  CDB204        0925          CALL    PUTREC          ;WHICH UPDATES HL
04A0  D1            0926          POP     DE              ;DE <- LAST ADDR TO DUMP
04A1  EB            0927          EX      DE,HL           ;HL <- LAST, DE <- CURRENT
04A2  18E5          0928          JR      SV1
                    0929
04A4  E1            0930  SV3:    POP     HL              ;END REC, BALANCE STACK
04A5  E1            0931          POP     HL              ;HL <- START ADDR
04A6  0600          0932          LD      B,0
04A8  1601          0933          LD      D,1             ;REC TYPE 1 (EOF)
04AA  CDB204        0934          CALL    PUTREC
04AD  CD9607        0935          CALL    SEC5            ;5 SEC DELAY
                    0936                                  ;STOP TAPE
04B0  F700          0937          EMT     0               ;BACK TO MONITOR
                    0938
                    0939  ;PUTREC - WRITE RECORD TO TAPE WITH CKSUM
                    0940  ; OUTPUT B BYTES FROM (HL), D=TYPE, C=CKSUM
```

```
                        0941
04B2    EF98            0942  PUTREC: CALR    RECNO           ;/DISPLAY ADDR (FROM HL)
04B4    3E4D            0943  PUTRNA: LD      A,SOH           ;PUT SOH (NOT IN CKSUM)
04B6    EF23            0944          CALR    TO
04B8    AF              0945          XOR     A               ;ZERO CKSUM
04B9    4F              0946          LD      C,A
04BA    78              0947          LD      A,B             ;PUT LENGTH
04BB    EF1E            0948          CALR    TO
04BD    7C              0949          LD      A,H             ;PUT ADDR H
04BE    EF1B            0950          CALR    TO
04C0    7D              0951          LD      A,L             ;PUT ADDR L
04C1    EF18            0952          CALR    TO
04C3    7A              0953          LD      A,D             ;PUT TYPE
04C4    EF15            0954          CALR    TO
04C6    78              0955          LD      A,B             ;CHECK LENGTH
04C7    B7              0956          OR      A
04C8    2806            0957          JR      Z,PR2           ;JUMP IF NO DATA
04CA    7E              0958  PR1:     LD      A,(HL)
04CB    23              0959          INC     HL
04CC    EF0D            0960          CALR    TO
04CE    10FA            0961          DJNZ    PR1
04D0    79              0962  PR2:    LD      A,C             ;GET CKSUM
04D1    ED44            0963          NEG
04D3    EF06            0964          CALR    TO              ;PUT CKSUM
04D5    0602            0965          LD      B,GAP
04D7    CD9807          0966          CALL    MS100           ;INTER RECORD GAP
04DA    C9              0967          RET
                        0968
                        0969  ;TO - PUT BYTE TO TAPE, FORM CKSUM IN C
                        0970
04DB    F703            0971  TO:      EMT     PUTBYT          ;A <- BYTE
04DD    81              0972          ADD     A,C             ;ADD CKSUM TO BYTE
04DE    4F              0973          LD      C,A             ;SAVE CKSUM
04DF    C9              0974          RET
                        0975
```

```
                        0976  *H GENERAL I/O ROUTINES
                        0977
                        0978  ;.KBDIN - READ KEYBOARD (CALLED VIA EMT)
                        0979  ;ON EXIT Z SET IF NO CHAR, ELSE CLEAR
                        0980  ;WATCH FOR CTRL C (RETURN TO MONITOR)
                        0981
04E0   E5               0982  .KBDIN: PUSH    HL
04E1   210340           0983          LD      HL,MASK
04E4   CBC6             0984          SET     CLRKBD,(HL)     ;SET UP TO CLEAR KBD
04E6   3AFC0F           0985          LD      A,(KBD)         ;A <- POSSIBLE CHAR
04E9   E67F             0986          AND     7FH             ;CLEAR MS BIT
04EB   2802             0987          JR      Z,KBD1          ;IF NO CHAR AVAILABLE
04ED   F712             0988          EMT     UPDATE          ;CLEAR KBD
04EF   CB86             0989  KBD1:   RES     CLRKBD,(HL)
04F1   F712             0990          EMT     UPDATE
04F3   E1               0991          POP     HL
04F4   C8               0992          RET     Z               ;NO CHAR AVAILABLE
04F5   FE03             0993          CP      CTRLC           ;IF NOT CTRL C
04F7   C0               0994          RET     NZ              ;RETURN THE CHAR
04F8   F700             0995          EMT     0               ;ELSE RESTART MONITOR
                        0996
                        0997  ;CLSOPN - CLOSE THEN OPEN (USED WHEN ROUTINE IS
                        0998  ;LONGER THAN FRAME BLANKING PERIOD)
                        0999
04FA   F70C             1000  CLSOPN: EMT     CLOSE           ;DROP THRO' TO OPNWT
                        1001
                        1002
                        1003  ;.OPNWT - OPEN VT MEMORY IN FRAME BLANKING PERIOD
                        1004
04FC   E5               1005  .OPNWT: PUSH    HL
04FD   F5               1006          PUSH    AF
04FE   21FE0F           1007          LD      HL,PORT1
0501   CB76             1008  OPNW1:  BIT     FRAME,(HL)      ;IF IN IT, WAIT
0503   20FC             1009          JR      NZ,OPNW1
0505   CB76             1010  OPNW2:  BIT     FRAME,(HL)      ;WAIT FOR NEXT
0507   28FC             1011          JR      Z,OPNW2
0509   210340           1012          LD      HL,MASK         ;OPEN MEMORY
050C   CBD6             1013          SET     VTOPN,(HL)
050E   F712             1014          EMT     UPDATE
0510   F1               1015          POP     AF
0511   E1               1016          POP     HL
0512   C9               1017          RET
                        1018
                        1019  ;.CLEAR - EXTERNAL CALL TO CLR WHICH DROPS
                        1020  ;THROUGH TO CLOSE
                        1021
0513   F70B             1022  .CLEAR: EMT     OPNWT
0515   CD2B05           1023          CALL    CLR
                        1024
                        1025  ;.CLOSE - CLOSE VT MEMORY
                        1026
0518   E5               1027  .CLOSE: PUSH    HL
0519   210340           1028          LD      HL,MASK
051C   CB96             1029          RES     VTOPN,(HL)
```

```
051E   F712       1030          EMT     UPDATE
0520   E1         1031          POP     HL
0521   C9         1032          RET
                  1033
                  1034  ;UPDATE - OUTPUT MASK TO PORT 0
                  1035
0522   F5         1036  .UPDAT: PUSH    AF
0523   3A0340     1037  UPD1:   LD      A,(MASK)
0526   32FC0F     1038          LD      (PORT0),A
0529   F1         1039          POP     AF
052A   C9         1040          RET
                  1041
                  1042  ;CLR - CLEAR THE VT SCREEN
                  1043  ;ON ENTRY A HOLDS THE NUMBER OF LINES TO
                  1044  ;CLEAR AND HL THE ADDRESS OF THE FIRST LINE
                  1045  ;ON EXIT HL POINTS TO ONE PAST LAST CHARACTER
                  1046
052B   111800     1047  CLR:    LD      DE,018H         ;INCR BETWEEN LINES
052E   0628       1048  CLR1:   LD      B,40            ;CHARS/LINE
0530   3680       1049  CLR2:   LD      (HL),BLANK      ;CLEAR A LINE
0532   23         1050          INC     HL
0533   10FB       1051          DJNZ    CLR2
0535   3D         1052          DEC     A               ;ALL LINES DONE
0536   C8         1053          RET     Z               ;YES
0537   19         1054          ADD     HL,DE           ;POINT TO NEXT LINE
0538   18F4       1055          JR      CLR1
                  1056
                  1057  ;AOUT, PMTOUT AND CROUT OUTPUT TO THE VT SCREEN
                  1058
053A   F5         1059  AOUT:   PUSH    AF              ;OUTPUT CR THEN A
053B   CD4505     1060          CALL    CROUT
053E   F1         1061          POP     AF
053F   1806       1062          JR      ..OUTC
                  1063
0541   3E3E       1064  PMTOUT: LD      A,'>'           ;OUTPUT PROMPT
0543   1802       1065          JR      ..OUTC          ;RETURNING VIA OUTC
                  1066
0545   3E0D       1067  CROUT:  LD      A,CR            ;RETURN VIA OUTC
0547               1068  ..OUTC:
                  1069
                  1070  ;.OUTC - PUTS ONE CHARACTER TO SCREEN
                  1071  ;CALLED VIA 'EMT OUTC' BY MONITOR
                  1072  ;INTERPRETS DELETE, CR, FORM FEED AND TAB
                  1073
0547   E5         1074  .OUTC:  PUSH    HL              ;SAVE ALL
0548   D5         1075          PUSH    DE
0549   C5         1076          PUSH    BC
054A   F5         1077          PUSH    AF
054B   F70B       1078          EMT     OPNWT           ;###
054D   2A0540     1079          LD      HL,(VTPTR)      ;HL <- CURRENT POS
0550   3680       1080          LD      (HL),BLANK      ;CURSOR OFF
0552   118A05     1081          LD      DE,SCRL         ;ELSE RET VIA SCRL
0555   D5         1082          PUSH    DE
                  1083
```

```
0556  FE7F      1084          CP      DELETE          ;IF DELETE
0558  2006      1085          JR      NZ,OUTC1
055A  3EC0      1086          LD      A,0C0H              ;ANY CHARS?
055C  BD        1087          CP      L
055D  D0        1088          RET     NC                  ;NO--IGNORE
055E  2B        1089          DEC     HL                  ;YES--DEC PTR
055F  C9        1090          RET
                1091
0560  FE20      1092  OUTC1:  CP      20H             ;ELSE IF NOT CONTR CHAR
0562  3808      1093          JR      C,OUTC2
0564  FE4F      1094          CP      'O'                 ;SWITCH 'O'
0566  2001      1095          JR      NZ,OUTC1A
0568  AF        1096          XOR     A
0569  77        1097  OUTC1A: LD      (HL),A              ;DISPLAY IT
056A  23        1098          INC     HL                  ;AND INC PTR
056B  C9        1099          RET
                1100
056C  D60D      1101  OUTC2:  SUB     CR              ;ELSE IF CARR RET
056E  2003      1102          JR      NZ,OUTC3
0570  2EE8      1103          LD      L,0E8H              ;FORCE A SCROLL
0572  C9        1104          RET
                1105
0573  3C        1106  OUTC3:  INC     A               ;ELSE IF FORM FEED
0574  2008      1107          JR      NZ,OUTC5
0576  3A0040    1108          LD      A,(NLINES)          ;CLEAR SCROLLING SCREEN
0579  2A0140    1109          LD      HL,(TOP)
057C  18AD      1110          JR      CLR
                1111
                1112  ; OUTC4:  COULD BE VERTICAL TAB
                1113
057E  C603      1114  OUTC5:  ADD     A,3             ;ELSE IF NOT HOR TAB
0580  C0        1115          RET     NZ                  ;IGNORE
0581  7D        1116          LD      A,L             ;GET LAST TAB POS
0582  E6F8      1117          AND     0F8H
0584  6F        1118          LD      L,A
0585  010800    1119          LD      BC,8            ;BUMP BY 8
0588  09        1120          ADD     HL,BC
0589  C9        1121          RET
                1122
                1123  ;SCRL - SCROLL THE VT SCREEN IF NECESSARY, CLEARING
                1124  ;LAST LINE, THEN RESTORE CURSOR AND UPDATE POINTER
                1125
058A  3EE8      1126  SCRL:   LD      A,0E8H
058C  BD        1127          CP      L               ;OFF THE END?
058D  202E      1128          JR      NZ,CURON        ;NO
                1129
058F  3A0040    1130          LD      A,(NLINES)
0592  3D        1131          DEC     A               ;A <- LINES TO SCROLL
0593  2A0140    1132          LD      HL,(TOP)
0596  EB        1133          EX      DE,HL           ;DE <- ADDR OF TOP LINE
0597  214000    1134          LD      HL,40H
059A  19        1135          ADD     HL,DE           ;HL <- ADDR OF NEXT LINE
059B  012800    1136  SCRL1:  LD      BC,40           ;BC <- LINE LENGTH
059E  EDB0      1137          LDIR
```

```
05A0   3D          1138          DEC     A
05A1   2810        1139          JR      Z,CLAST         ;DONE?
05A3   011800      1140          LD      BC,18H          ;NO
05A6   09          1141          ADD     HL,BC           ;ADVANCE POINTERS TO NEXT
05A7   EB          1142          EX      DE,HL
05A8   09          1143          ADD     HL,BC
05A9   EB          1144          EX      DE,HL
05AA   F5          1145          PUSH    AF
05AB   E607        1146          AND     7               ;###PAUSE EVERY 8
05AD   CCFA04      1147          CALL    Z,CLSOPN        ;###
05B0   F1          1148          POP     AF
05B1   18E8        1149          JR      SCRL1
                   1150
05B3   3EC0        1151  CLAST:  LD      A,0C0H          ;A <- START OF LAST
05B5   BD          1152          CP      L               ;DONE?
05B6   2805        1153          JR      Z,CURON         ;YES
05B8   2B          1154          DEC     HL
05B9   3680        1155          LD      (HL),BLANK
05BB   18F8        1156          JR      CLAST+2
                   1157
05BD   367F        1158  CURON:  LD      (HL),CURSOR
05BF   220540      1159          LD      (VTPTR),HL
05C2   F70C        1160          EMT     CLOSE           ;###
05C4   F1          1161  RET1:   POP     AF              ;STANDARD EXITS
05C5   C1          1162  RET2:   POP     BC
05C6   D1          1163          POP     DE
05C7   E1          1164          POP     HL
05C8   C9          1165          RET
                   1166
                   1167  ;MOUT - OUTPUT CR, THEN STRING, USING .MSG
                   1168  ;  REG A LOST, OTHERS PRESERVED
                   1169
05C9   211300      1170  ERROUT: LD      HL,MSGE
                   1171
05CC   CD4505      1172  MOUT:   CALL    CROUT
                   1173
                   1174  ;.MSG - FAST STRING OUTPUT TO VT SCROLLER
                   1175  ;  ENTER WITH (HL) -> STRING
                   1176  ;  END OF STRING (EOS) IS A -VE BYTE
                   1177  ;  ALL REGISTERS PRESERVED
                   1178
05CF   E5          1179  .MSG:   PUSH    HL              ;STANDARD SAVE SEQUENCE
05D0   D5          1180          PUSH    DE
05D1   C5          1181          PUSH    BC
05D2   F5          1182          PUSH    AF
05D3   ED5B0540    1183          LD      DE,(VTPTR)
05D7   F70B        1184          EMT     OPNWT           ;###
05D9   EDA0        1185  .MSG1:  LDI                     ;CAN ALWAYS MOVE ONE CHAR
05DB   CB7E        1186          BIT     7,(HL)          ;REACHED END OF STRING?
05DD   28FA        1187          JR      Z,.MSG1         ;CONTINUE IF NOT
05DF   EB          1188          EX      DE,HL           ;USE SCROLLER TO REPLACE
05E0   18A8        1189          JR      SCRL            ;CURSOR, DO CR IF NEEDED
                   1190
                   1191  ;FSTLST - PROMPT FOR 'FIRST' ADDR AND 'LAST' ADDR
```

```
                        1192  ;ON RETURN, DE = FIRST, BC = LAST AND HL = (LAST - FIRST)
                        1193  ;ERROR IF FIRST > LAST
                        1194
05E2   211B00           1195  FSTLST: LD      HL,MSGF         ;GET 'FIRST'
05E5   EF11             1196          CALR    MGTHEX
05E7   EB               1197          EX      DE,HL           ;DE <- FIRST
05E8   212B00           1198          LD      HL,MSGL         ;GET 'LAST'
05EB   EF0B             1199          CALR    MGTHEX
05ED   44               1200          LD      B,H             ;BC <- LAST
05EE   4D               1201          LD      C,L
05EF   B7               1202          OR      A               ;TEST IF VALID, CLR CY
05F0   ED52             1203          SBC     HL,DE
05F2   D0               1204          RET     NC              ;WITH HL = LAST - FIRST
                        1205
                        1206  ;ERROR - ISSUE ERROR MESSAGE AND RESTART
                        1207
05F3   CDC905           1208  ERROR:  CALL    ERROUT
05F6   F700             1209          EMT     0               ;AND RESET STACK
                        1210
                        1211  ;GETHEX - GETS HEX VALUE FROM KEYBOARD
                        1212  ;ON RETURN HL CONTAINS VALUE, C CONTAINS NUMBER
                        1213  ;OF HEX CHARS, B CONTAINS TERMINATOR AND DE IS
                        1214  ;UNCHANGED.  INTERPRETS DELETE.
                        1215
05F8   CDCC05           1216  MGTHEX: CALL    MOUT            ;OUTPUT CR, THEN STR >
05FB   1803             1217          JR      PGTHEX
                        1218
05FD   CD4505           1219  CGTHEX: CALL    CROUT           ;OUTPUT CR, >
                        1220
0600   CD4105           1221  PGTHEX: CALL    PMTOUT          ;OUTPUT PROMPT
                        1222
0603   210000           1223  .GTHEX: LD      HL,0            ;CLEAR BUFFER AND
0606   4D               1224          LD      C,L             ;CHAR COUNTER
0607   F702             1225  NXTC:   EMT     KBDIN           ;GET A CHAR
0609   28FC             1226          JR      Z,NXTC
060B   47               1227          LD      B,A             ;SAVE LAST IN B
060C   FE30             1228          CP      '0'             ;CONTROL CHAR?
060E   D8               1229          RET     C               ;YES--RETURN
060F   FE7F             1230          CP      DELETE          ;IF DELETE
0611   281E             1231          JR      Z,NXTC2         ;DELETE LAST DIG (IF ANY)
0613   F716             1232          EMT     OUTC            ;ECHO PRINTING CHAR
0615   D630             1233          SUB     '0'             ;CONVERT TO HEX
0617   FE0A             1234          CP      10              ;0 TO 9?
0619   3808             1235          JR      C,NXTC1         ;YES
061B   D607             1236          SUB     7
061D   FE0A             1237          CP      0AH             ;LESS THAN A?
061F   D8               1238          RET     C               ;YES--RETURN
0620   FE10             1239          CP      10H             ;MORE THAN F?
0622   D0               1240          RET     NC              ;YES
                        1241
0623   29               1242  NXTC1:  ADD     HL,HL           ;SHIFT LEFT 4
0624   29               1243          ADD     HL,HL
0625   29               1244          ADD     HL,HL
0626   29               1245          ADD     HL,HL
```

```
0627  B5        1246          OR    L              ;MERGE NEW
0628  6F        1247          LD    L,A
0629  3E03      1248          LD    A,3            ;HAVE THERE BEEN 4 CHARS?
062B  B9        1249          CP    C
062C  3812      1250          JR    C,NXTC4        ;IF C > 3
062E  0C        1251          INC   C              ;ELSE BUMP CHAR COUNT
062F  18D6      1252          JR    NXTC
                1253
0631  0C        1254  NXTC2:  INC   C              ;DELETE, TEST IF ANY CHARS
0632  0D        1255          DEC   C
0633  28D2      1256          JR    Z,NXTC         ;Z IF NONE
0635  0604      1257          LD    B,4            ;SHIFT RIGHT 4
0637  CB3C      1258  NXTC3:  SRL   H
0639  CB1D      1259          RR    L
063B  10FA      1260          DJNZ  NXTC3
063D  0D        1261          DEC   C              ;DEC COUNT
063E  180E      1262          JR    NXTC5          ;ECHO DELETE
                1263
0640  D5        1264  NXTC4:  PUSH  DE             ;5 CHARS, SAVE DE
0641  EB        1265          EX    DE,HL          ;DE <- HEX VALUE
0642  2A0540    1266          LD    HL,(VTPTR)     ;HL <- ADDR OF CURSOR
0645  7D        1267          LD    A,L
0646  D605      1268          SUB   5
0648  6F        1269          LD    L,A            ;HL <- ADDR OF START OF STR
0649  CD8806    1270          CALL  DEOUTW         ;DISPLAY TRUE 4 CHAR STR
064C  EB        1271          EX    DE,HL          ;HL <- HEX VALUE
064D  D1        1272          POP   DE             ;RESTORE DE
                1273
064E  3E7F      1274  NXTC5:  LD    A,DELETE       ;DELETE FIFTH CHAR
0650  F716      1275          EMT   OUTC
0652  18B3      1276          JR    NXTC
                1277
                1278  ;GETNAM - GET NAME FROM KBD, HANDLE DELETE
                1279
0654  21D107    1280  GETNAM: LD    HL,MSGN        ;OUTPUT CR NAME>
0657  CDCC05    1281          CALL  MOUT
065A  CD4105    1282          CALL  PMTOUT
065D  0606      1283          LD    B,6
065F  214540    1284          LD    HL,NAME        ;HL <- BUFFER ADDR
0662  CDCD00    1285  NM1:    CALL  GETC           ;GET A CHAR
0665  FE30      1286          CP    '0'
0667  380A      1287          JR    C,NM2
0669  FE7F      1288          CP    DELETE
066B  280E      1289          JR    Z,NM4
066D  F716      1290          EMT   OUTC           ;ECHO CHAR
066F  77        1291          LD    (HL),A         ;SAVE IT IN BUF
0670  23        1292          INC   HL
0671  10EF      1293          DJNZ  NM1            ;GET UP TO 6 CHARS
0673  04        1294  NM2:    INC   B              ;WHILE (B != 0)
0674  05        1295  NM3:    DEC   B
0675  C8        1296          RET   Z
0676  3620      1297          LD    (HL),' '       ;PAD WITH SPACE (NOT GRAPHICS BLANK)
0678  23        1298          INC   HL
0679  18F9      1299          JR    NM3
```

```
067B  78        1300  NM4:    LD      A,B             ;TEST IF ANY CHARS
067C  FE06      1301          CP      6
067E  28E2      1302          JR      Z,NM1           ;Z IF NONE
0680  04        1303          INC     B               ;ELSE UPDATE CNTR
0681  2B        1304          DEC     HL              ;AND DELETE CHAR FROM BUF
0682  3E7F      1305          LD      A,DELETE        ;ECHO DELETE
0684  F716      1306          EMT     OUTC
0686  18DA      1307          JR      NM1
                1308
```

```
                        1309  *H VT OUTPUT ROUTINES
                        1310  ;HL IS SCREEN ADDRESS
                        1311
                        1312  ;DEOUTW - DEOUT WHEN VT IS CLOSED
                        1313  ;DEOUTC - DEOUT, THEN CLOSE
                        1314
0688    F70B            1315  DEOUTW: EMT     OPNWT           ;###
068A    F714            1316  DEOUTC: EMT     DEOUT
068C    F70C            1317          EMT     CLOSE           ;###
068E    C9              1318          RET
                        1319
                        1320  ;.DEOUT - PUTS 4 HEX CHARS FROM DE
                        1321
068F    7A              1322  .DEOUT: LD      A,D
0690    CD9406          1323          CALL    .BYTEO
0693    7B              1324          LD      A,E
                        1325
                        1326  ;.BYTEO - PUTS 2 HEX CHARS FROM A
                        1327
0694    F5              1328  .BYTEO: PUSH    AF
0695    0F              1329          RRCA                    ;GET UPPER 4 BITS
0696    0F              1330          RRCA
0697    0F              1331          RRCA
0698    0F              1332          RRCA
0699    CD9D06          1333          CALL    HEXO
069C    F1              1334          POP     AF
                        1335
                        1336  ;HEXO - PUTS LOWER 4 BITS OF A IN HEX
                        1337
069D    E60F            1338  HEXO:   AND     0FH             ;MASK TO 4 BITS
069F    C690            1339          ADD     A,90H           ;CONVERT TO HEX
06A1    27              1340          DAA
06A2    CE40            1341          ADC     A,40H
06A4    27              1342          DAA
06A5    77              1343  CHO:    LD      (HL),A          ;PUT OUT CHARACTER
06A6    23              1344          INC     HL              ;BUMP POINTER
06A7    C9              1345          RET
                        1346
```

```
                        1347  *H TAPE I/O ROUTINES
                        1348
                        1349  ;.GTBYT - GETS NEXT BYTE FROM TAPE INTO A
                        1350  ;CALLED VIA EMT
                        1351
06A8   E5               1352  .GTBYT: PUSH    HL              ;SAVE ALL BUT A
06A9   D5               1353          PUSH    DE
06AA   C5               1354          PUSH    BC
06AB   CDF606           1355          CALL    GETCYC
06AE   AF               1356          XOR     A
06AF   57               1357          LD      D,A
06B0   3D               1358          DEC     A
06B1   5F               1359          LD      E,A
                        1360
06B2   CB5B             1361  GB1:    BIT     3,E
06B4   2001             1362          JR      NZ,GB2
06B6   15               1363          DEC     D
06B7   CDF606           1364  GB2:    CALL    GETCYC
06BA   3801             1365          JR      C,GB3
06BC   14               1366          INC     D
06BD   CB13             1367  GB3:    RL      E
06BF   3E01             1368          LD      A,1
06C1   BA               1369          CP      D
06C2   30EE             1370          JR      NC,GB1
                        1371
06C4   CB5B             1372          BIT     3,E
06C6   280C             1373          JR      Z,GB4
06C8   CDF606           1374          CALL    GETCYC
06CB   CB13             1375          RL      E
06CD   CB5B             1376          BIT     3,E
06CF   2803             1377          JR      Z,GB4
06D1   CDF606           1378          CALL    GETCYC
06D4   0608             1379  GB4:    LD      B,8
06D6   50               1380          LD      D,B
                        1381
06D7   1E00             1382  GB5:    LD      E,0
06D9   3EF8             1383          LD      A,-8
06DB   82               1384          ADD     A,D
06DC   57               1385          LD      D,A
                        1386
06DD   CDF606           1387  GB6:    CALL    GETCYC
06E0   14               1388          INC     D
06E1   1C               1389          INC     E
06E2   3802             1390          JR      C,GB7
06E4   14               1391          INC     D
06E5   1D               1392          DEC     E
06E6   3E07             1393  GB7:    LD      A,7
06E8   BA               1394          CP      D
06E9   30F2             1395          JR      NC,GB6
                        1396
06EB   3E03             1397          LD      A,3
06ED   BB               1398          CP      E
06EE   CB19             1399          RR      C
06F0   10E5             1400          DJNZ    GB5
```

```
                    1401
06F2    79          1402            LD      A,C
06F3    C3C505      1403            JP      RET2            ;STANDARD EXIT
                    1404
                    1405    ;GETCYC - TIME A CYCLE FROM TAPE
                    1406    ;/ALLOW ESCAPE BY CTRL C
                    1407
06F6    21FE0F      1408    GETCYC: LD      HL,PORT1
06F9    CDE004      1409    GC1:    CALL    .KBDIN          ;CHECK FOR CTRL C
06FC    CB6E        1410            BIT     TAPE,(HL)
06FE    28F9        1411            JR      Z,GC1
0700    CB6E        1412            BIT     TAPE,(HL)
0702    28F5        1413            JR      Z,GC1
0704    CB6E        1414    GC2:    BIT     TAPE,(HL)
0706    20FC        1415            JR      NZ,GC2
0708    CB6E        1416            BIT     TAPE,(HL)
070A    20F8        1417            JR      NZ,GC2
070C    CD7307      1418            CALL    RDCNT
070F    CD8607      1419            CALL    RESCNT
0712    FE47        1420            CP      VALUE
0714    C9          1421            RET
                    1422
                    1423    ;GETSYN - WAIT FOR START OF RECORD (TRAP ROUTINE)
                    1424    ;CALLED WITH SOH IN A. WAITS FOR INTERRECORD GAP,
                    1425    ;THEN SOH CHAR.  REGS UNCHANGED.
                    1426
0715    E5          1427    .GTSYN: PUSH    HL              ;STANDARD SAVE
0716    D5          1428            PUSH    DE
0717    C5          1429            PUSH    BC
0718    F5          1430            PUSH    AF
0719    4F          1431            LD      C,A             ;C <- SOH
071A    CD0D40      1432    GTSN1:  CALL    GETGAP          ;WAIT FOR GAP
071D    F704        1433            EMT     GETBYT          ;READ FIRST CHAR
071F    B9          1434            CP      C               ;IS IT CORRECT?
0720    20F8        1435            JR      NZ,GTSN1        ;NO--ASSUME A GLITCH
0722    181F        1436            JR      PBY2            ;STANDARD EXIT
                    1437
                    1438    ;.GTGAP - FIND INTER RECORD GAP (GAPLEN '1' CYCLES)
                    1439    ;CALLED VIA GETGAP (RAM VECTOR) SO THAT IT CAN BE
                    1440    ;SKIPPED FOR PAPER TAPE
                    1441
0724    06F0        1442    .GTGAP: LD      B,GAPLEN-256    ;SET B TO # OF '1' CYCLES
0726    CDF606      1443    GG1:    CALL    GETCYC
0729    30F9        1444            JR      NC,.GTGAP       ;RESET B IF A '0' CYCLE SEEN
072B    10F9        1445            DJNZ    GG1             ;COUNT DOWN B '1' CYCLES
072D    C9          1446            RET
                    1447
```

```
                        1448  *E
                        1449  ;.PTBYT - OUTPUTS BYTE TO TAPE FROM A
                        1450  ;CALLED VIA EMT
                        1451
072E   E5               1452  .PTBYT: PUSH    HL              ;SAVE ALL
072F   D5               1453          PUSH    DE
0730   C5               1454          PUSH    BC
0731   F5               1455          PUSH    AF
0732   2F               1456          CPL
0733   4F               1457          LD      C,A
0734   3E01             1458          LD      A,1
0736   EF0E             1459          CALR    PUTBIT
0738   060A             1460          LD      B,10
                        1461
073A   CB39             1462  PBY1:   SRL     C
073C   17               1463          RLA
073D   EF07             1464          CALR    PUTBIT
073F   10F9             1465          DJNZ    PBY1
                        1466
0741   F702             1467          EMT     KBDIN
0743   C3C405           1468  PBY2:   JP      RET1            ;STANDARD EXIT
                        1469
                        1470  ;PUTBIT - PUT BIT 0 OF A TO TAPE
                        1471
0746   C5               1472  PUTBIT: PUSH    BC
0747   210340           1473          LD      HL,MASK
074A   0604             1474          LD      B,4
074C   E603             1475          AND     11B
074E   281F             1476          JR      Z,PBT4
0750   FE03             1477          CP      11B
0752   281B             1478          JR      Z,PBT4
0754   CBA6             1479          RES     FREQ,(HL)
0756   FE01             1480          CP      01B
0758   2802             1481          JR      Z,PBT1
075A   CBE6             1482          SET     FREQ,(HL)
                        1483
075C   F5               1484  PBT1:   PUSH    AF
075D   7E               1485          LD      A,(HL)
075E   11FC0F           1486          LD      DE,PORT0
0761   21FE0F           1487          LD      HL,PORT1
0764   CB4E             1488  PBT2:   BIT     CLK,(HL)
0766   28FC             1489          JR      Z,PBT2
0768   CB4E             1490  PBT3:   BIT     CLK,(HL)
076A   20FC             1491          JR      NZ,PBT3
076C   12               1492          LD      (DE),A
076D   F1               1493          POP     AF
076E   05               1494          DEC     B
                        1495
076F   F710             1496  PBT4:   EMT     EDGE
0771   C1               1497          POP     BC
0772   C9               1498          RET
                        1499
```

```
                        1500  *E
                        1501  ;RDCNT - READ THE 8 USEC COUNTER INTO A (REGS PRESERVED)
                        1502
0773    E5              1503  RDCNT:  PUSH    HL
0774    C5              1504          PUSH    BC
0775    21FD0F          1505          LD      HL,CNTR
0778    46              1506          LD      B,(HL)
0779    7E              1507          LD      A,(HL)
077A    4E              1508          LD      C,(HL)
077B    66              1509          LD      H,(HL)
077C    B8              1510          CP      B
077D    2804            1511          JR      Z,RC1
077F    B9              1512          CP      C
0780    2801            1513          JR      Z,RC1
0782    7C              1514          LD      A,H
0783    C1              1515  RC1:    POP     BC
0784    E1              1516          POP     HL
0785    C9              1517          RET
                        1518
                        1519  ;RESCNT - CLEAR THE 8 USEC COUNTER (REGS PRESERVED)
                        1520  ;(N.B. TO RETAIN TIMING, DOES NOT USE TRAP FOR 'UPDATE')
                        1521
0786    E5              1522  RESCNT: PUSH    HL
0787    210340          1523          LD      HL,MASK
078A    CBF6            1524          SET     CLRCNT,(HL)
078C    CD2205          1525          CALL    .UPDAT
078F    CBB6            1526          RES     CLRCNT,(HL)
0791    CD2205          1527          CALL    .UPDAT
0794    E1              1528          POP     HL
0795    C9              1529          RET
                        1530
                        1531  ;SEC5 - WAIT 5 SEC (DESTROYS B)
                        1532
0796    0632            1533  SEC5:   LD      B,50            ;SET 50 * 100 MSEC
                        1534
                        1535  ;MS100 - WAIT FOR B 100 MSEC INTERVALS
                        1536
0798    C5              1537  MS100:  PUSH    BC
0799    0678            1538          LD      B,120           ;SET 120 * 1/1200 SEC
079B    F710            1539          EMT     EDGE
079D    C1              1540          POP     BC
079E    10F8            1541          DJNZ    MS100
07A0    C9              1542          RET
                        1543
                        1544  ;EDGE - WAIT FOR B 1200 HZ EDGES
                        1545  ;       (ALSO USED AS 833 1/3 USEC TIMER)
                        1546
07A1    E5              1547  .EDGE:  PUSH    HL
07A2    D5              1548          PUSH    DE
07A3    C5              1549          PUSH    BC
07A4    F5              1550          PUSH    AF
07A5    21FE0F          1551          LD      HL,PORT1
07A8    CB4E            1552  ED1:    BIT     CLK,(HL)        ;WAIT FOR CLOCK HIGH
07AA    28FC            1553          JR      Z,ED1
```

```
07AC   CB4E      1554  ED2:   BIT     CLK,(HL)        ;WAIT FOR CLOCK LOW
07AE   20FC      1555         JR      NZ,ED2
07B0   10F6      1556         DJNZ    ED1
07B2   C3C405    1557         JP      RET1
                 1558
```

```
                        1559  *H ADDRESSES AND PARAMETERS
                        1560
                        1561  ;DISPATCH TABLE FOR TRAP CODES 11 TO 24
                        1562
        000B            1563  OPNWT   EQU     11
        000C            1564  CLOSE   EQU     12
        000D            1565  GRAFIX  EQU     13
        000E            1566  SCROLL  EQU     14
        000F            1567  CLEAR   EQU     15
        0010            1568  EDGE    EQU     16
        0011            1569  GETSYN  EQU     17
        0012            1570  UPDATE  EQU     18
        0013            1571  GETHEX  EQU     19
        0014            1572  DEOUT   EQU     20
        0015            1573  BYTEO   EQU     21
        0016            1574  OUTC    EQU     22
        0017            1575  MSG     EQU     23
        0018            1576  CHAN    EQU     24
                        1577
07B5    FC04            1578  TRTBL:  DEFW    .OPNWT
07B7    1805            1579          DEFW    .CLOSE
07B9    2603            1580          DEFW    .GRAFX
07BB    1A03            1581          DEFW    .SCROL
07BD    1305            1582          DEFW    .CLEAR
07BF    A107            1583          DEFW    .EDGE
07C1    1507            1584          DEFW    .GTSYN
07C3    2205            1585          DEFW    .UPDAT
07C5    0306            1586          DEFW    .GTHEX
07C7    8F06            1587          DEFW    .DEOUT
07C9    9406            1588          DEFW    .BYTEO
07CB    4705            1589          DEFW    .OUTC
07CD    CF05            1590          DEFW    .MSG
07CF    6E00            1591          DEFW    .CHAN
                        1592
                        1593  ;SYSTEM MESSAGES - THE MAJORITY OF THESE ARE INTERSPERSED
                        1594  ;WITH THE JUMPS TO THE 'RST' VECTORS
                        1595
07D1    4E414D45        1596  MSGN:   DEFM    'NAME'
07D5    FF              1597          DEFB    -1
07D6    434F5320        1598  MSGV:   DEFM    'COS 2.3'       ;VERSION NUMBER
07DD    FF              1599          DEFB    -1
                        1600
                        1601  ;INITBK - THIS THE INITIAL STATE OF RAM AND
                        1602  ;IS SET UP BY BEGIN ON RESET OR PARTIALLY BY CTRL C
                        1603
07DE    18              1604  INITBK: DEFB    24              ;NLINES
07DF    0008            1605          DEFW    0800H           ;TOP
07E1    92              1606          DEFB    MKINIT          ;MASK
        0004            1607  CCLEN   EQU     $-INITBK
07E2    00              1608          DEFB    0               ;SSFLG
07E3    C00D            1609          DEFW    0DC0H           ;VTPTR
07E5    0041            1610          DEFW    STACK           ;MPTR
07E7    10              1611          DEFB    16              ;RPTR
07E8    00              1612          DEFB    0               ;RSET
```

```
07E9  0000      1613          DEFW    0               ;ADDR
07EB  C32407    1614          JP      .GTGAP          ;GETGAP.
07EE  C34705    1615          JP      .OUTC
07F1  C3E004    1616          JP      .KBDIN          ;KBDIN
07F4  C32E07    1617          JP      .PTBYT          ;PUTBYT
07F7  C3A806    1618          JP      .GTBYT          ;GETBYT
07FA  C31040    1619          JP      VTV             ;LP VECTOR
      001F      1620  INITLN  EQU     $-INITBK
                1621
07FD  04        1622  GRAFBK: DEFB    4               ;NLINES
07FE  000D      1623          DEFW    0D00H           ;TOP
                1624
      07FF      1625  MONEND  EQU     $-1
      0800      1626  MONLEN  EQU     MONEND-MONST+1
                1627
                1628  ;DEDICATED RAM FOR SYSTEM
                1629
0800            1630          ORG     RAM
                1631
4000            1632  NLINES: DEFS    1               ;LINES TO SCROLL
4001            1633  TOP:    DEFS    2               ;TOP OF SCROLLED AREA
4003            1634  MASK:   DEFS    1               ;PORT 0 MASK
4004            1635  SSFLG:  DEFS    1               ;SINGLE STEP FLAG
4005            1636  VTPTR:  DEFS    2               ;VT POINTER
4007            1637  MPTR:   DEFS    2               ;MEMORY POINTER
4009            1638  RPTR:   DEFS    1               ;REGISTER POINTER
400A            1639  RSET:   DEFS    1               ;STANDARD/ALT REG SWITCH
400B            1640  ADDR:   DEFS    2               ;USER COLD START ADDR
400D            1641  GETGAP: DEFS    3               ;GAP VECTOR (ALTER FOR PAPER TAPE)
                1642
                1643  ;TRANSFER VECTORS
                1644
      0002      1645  KBDIN   EQU     2
      0003      1646  PUTBYT  EQU     3
      0004      1647  GETBYT  EQU     4
                1648
4010            1649  VTV:    DEFS    3               ;CONSOLE OUT
4013            1650  KBDV:   DEFS    3               ;CONSOLE IN
4016            1651  TOV:    DEFS    3               ;TAPE OUT
4019            1652  TIV:    DEFS    3               ;TAPE IN
401C            1653  LPV:    DEFS    3               ;LINE PRINTER OUT
                1654
                1655                                  ;HERE DOWN INITIALISED TO 'OFFH'
401F            1656  OUT1:   DEFS    3               ;AUX I/O, USED BY FILE SYSTEMS
4022            1657  OUT2:   DEFS    3
4025            1658  IN1:    DEFS    3
4028            1659  IN2:    DEFS    3
402B            1660  IN3:    DEFS    3
                1661
402E            1662  RST8:   DEFS    3               ;RESTART VECTORS
4031            1663  RST10:  DEFS    3
4034            1664  RST18:  DEFS    3
4037            1665  RST20:  DEFS    3
                1666
```

```
403A              1667  NMIX:   DEFS    3               ;NMI EXTENSION VECTOR
403D              1668  TRAPX:  DEFS    3               ;TRAP EXTENSION VECTOR
      0021        1669  FFLEN   EQU     $-OUT1
4040              1670  MONX:   DEFS    3               ;MONITOR EXTENSION VECTOR
4043              1671  HIMEM:  DEFS    2               ;TOP OF RAM
                  1672
                  1673  ;GENERAL STORAGE (NOT INITIALISED)
                  1674
4045              1675  NAME:   DEFS    6               ;TARGET FILE NAME
404B              1676  PATL:   DEFS    1               ;LENGTH OF PATTERN
404C              1677  PATN:   DEFS    1               ;START OF PATTERN
                  1678
      4100        1679  STACK   EQU     RAM+100H
                  1680
                  1681  ;PORT ADDRESSES
                  1682
      0FFC        1683  KBD     EQU     0FFCH
      0FFC        1684  PORT0   EQU     0FFCH
      0FFD        1685  CNTR    EQU     0FFDH
      0FFE        1686  PORT1   EQU     0FFEH
                  1687
                  1688  ;PORT 0 BIT ALLOCATION
                  1689
      0092        1690  MKINIT  EQU     10010010B       ;INITIAL STATE OF PORT 0 MASK
      0007        1691  PAGE    EQU     7               ;MEMORY PAGE SWITCH (=1 FOR OFF)
      0006        1692  CLRCNT  EQU     6               ;CLEAR 8 USEC COUNTER
      0005        1693  RLY2    EQU     5               ;RELAY 2 (=0 FOR CLOSED)
      0004        1694  FREQ    EQU     4               ;SET 2400 HZ
      0003        1695  RLY1    EQU     3               ;RELAY 1 (=0 FOR CLOSED)
      0002        1696  VTOPN   EQU     2               ;OPEN VT MEMORY
      0001        1697  SINGLE  EQU     1               ;ENABLE SINGLE STEP (=0 FOR SET)
      0000        1698  CLRKBD  EQU     0               ;CLEAR KEYBOARD LATCH
                  1699
                  1700  ;PORT 1 BIT ALLOCATION
                  1701
      0007        1702  LINE    EQU     7               ;VT LINE WAVE FORM
      0006        1703  FRAME   EQU     6               ;VT FRAME WAVE FORM
      0005        1704  TAPE    EQU     5               ;CASSETTE SIGNAL SENSE
      0003        1705  VOL     EQU     3               ;CASSETTE VOLUME SENSE
      0002        1706  RESET   EQU     2               ;RESET BUTTON
      0001        1707  CLK     EQU     1               ;1200 HZ CLOCK
                  1708
      0047        1709  VALUE   EQU     47H             ;TAPE CYCLE DISCRIMINATOR
      0080        1710  RECLEN  EQU     128             ;RECORD LENGTH IN BYTES
      0002        1711  GAP     EQU     2               ;INTER REC GAP (100'S OF MSEC)
      00F0        1712  GAPLEN  EQU     240             ;GAP TO FIND WHEN READING ('1' CYCLES)
      004D        1713  SOH     EQU     01001101B       ;START OF RECORD (4DH)
                  1714
      000A        1715  LF      EQU     0AH
      000C        1716  FORM    EQU     0CH
      000D        1717  CR      EQU     0DH
                  1718
      0003        1719  CTRLC   EQU     3
      0015        1720  CTRLU   EQU     15H
```

```
001A        1721  CTRLZ   EQU     1AH
            1722
0080        1723  BLANK   EQU     80H             ;GRAPHICS BLANK
005B        1724  LARR    EQU     5BH
005D        1725  RARR    EQU     5DH
005E        1726  UPARR   EQU     5EH
007F        1727  DELETE  EQU     7FH
007F        1728  CURSOR  EQU     7FH
            1729
```

```
                1730  *H SYMBOL TABLE
0000            1731        END

SYMBOL TABLE:

...OUTC  0547    .BYTEO  0694    .CHAN   006E    .CHN1   0075
.CHN2   0076    .CLEAR  0513    .CLOSE  0518    .DEOUT  068F
.EDGE   07A1    .GRAFX  0326    .GTBYT  06A8    .GTGAP  0724
.GTHEX  0603    .GTSYN  0715    .KBDIN  04E0    .MSG    05CF
.MSG1   05D9    .OPNWT  04FC    .OUTC   0547    .PTBYT  072E
.SCROL  031A    .UPDAT  0522    ADDR    400B    AOUT    053A
BEGIN   0337    BG1     0347    BG2     0358    BLANK   0080
BREAK   0038    BYTEO   0015    CCLEN   0004    CGTHEX  05FD
CHAN    0010    CHO     06A5    CLAST   05B3    CLEAR   000F
CLK     0001    CLOSE   000C    CLR     052B    CLR1    052E
CLR2    0530    CLRCNT  0006    CLRKBD  0000    CLSOPN  04FA
CMD     01CD    CMD1    01E5    CMD2    01FE    CNTR    0FFD
COMP    03F2    CONTC   0371    CONTIN  0289    CR      000D
CROUT   0545    CTRLC   0003    CTRLU   0015    CTRLZ   001A
CURON   05BD    CURSOR  007F    DELETE  007F    DEOUT   0014
DEOUTC  068A    DEOUTW  0688    ED1     07A8    ED2     07AC
EDGE    0010    ERROR   05F3    ERROUT  05C9    FFLEN   0021
FIND    03CF    FORM    000C    FPJMP   0313    FPM     00D6
FPM1    00DF    FPM2    00E3    FRAME   0006    FREQ    0004
FSTLST  05E2    GAP     0002    GAPLEN  00F0    GB1     06B2
GB2     06B7    GB3     06BD    GB4     06D4    GB5     06D7
GB6     06DD    GB7     06E6    GC1     06F9    GC2     0704
GET     0438    GETBYT  0004    GETC    00CD    GETCYC  06F6
GETGAP  400D    GETHDR  0429    GETHEX  0013    GETNAM  0654
GETPAT  02D7    GETREC  03FA    GETSYN  0011    GG1     0726
GP1     02DB    GP2     02EE    GP3     0307    GPNXT   02EE
GR1     0418    GR2     0426    GRAFBK  07FD    GRAFIX  000D
GRAFNC  032D    GT1     0442    GTSN1   071A    HEXCAL  02BB
HEXO    069D    HIMEM   4043    IN1     4025    IN2     4028
IN3     402B    IND     0255    INIT    031D    INITBK  07DE
INITLN  001F    IT1     0320    JUMP    0316    KBD     0FFC
KBD1    04EF    KBDIN   0002    KBDV    4013    LARR    005B
LDER1   040C    LDERR   0409    LDRET   0092    LF      000A
LINE    0007    LOAD    03C9    LOOKUP  0227    LPV     401C
MASK    4003    MD1     01A8    MD2     01B1    MDIS    0198
MGTHEX  05F8    MKINIT  0092    MONEND  07FF    MONLEN  0800
MONST   0000    MONX    4040    MOUT    05CC    MOVE    03A5
MOVER   0335    MPTR    4007    MS100   0798    MSG     0017
MSGB    000B    MSGE    0013    MSGF    001B    MSGL    002B
MSGM    0006    MSGN    07D1    MSGP    0034    MSGS    0023
MSGV    07D6    MV1     03C3    MV2     03C6    NAME    4045
NLINES  4000    NM1     0662    NM2     0673    NM3     0674
NM4     067B    NMI     00A2    NMI1    00AF    NMI1A   00B7
NMI2    00B8    NMIX    403A    NXTC    0607    NXTC1   0623
NXTC2   0631    NXTC3   0637    NXTC4   0640    NXTC5   064E
NXTROM  1000    OPNW1   0501    OPNW2   0505    OPNWT   000B
OUT1    401F    OUT2    4022    OUTC    0016    OUTC1   0560
OUTC1A  0569    OUTC2   056C    OUTC3   0573    OUTC5   057E
PAGE    0007    PATL    404B    PATN    404C    PBT1    075C
```

PBT2	0764	PBT3	0768	PBT4	076F	PBY1	073A
PBY2	0743	PGTHEX	0600	PMTOUT	0541	PORTO	0FFC
PORT1	0FFE	PR1	04CA	PR2	04D0	PT1	02A5
PT2	02B8	PUT	0297	PUTBIT	0746	PUTBYT	0003
PUTREC	04B2	PUTRNA	04B4	RA1	026D	RAM	4000
RARR	005D	RC1	0783	RCALL	0085	RD1	0118
RD2	0124	RD3	013E	RD4	0153	RD5	015B
RD6	0165	RDCNT	0773	RDIS	00F3	RECLEN	0080
RECNO	044C	REL	025E	RESCNT	0786	RESET	0002
RET1	05C4	RET2	05C5	RLY1	0003	RLY2	0005
RN1	017C	RNAM	0174	RNSTR	0180	RNTR	039E
RPTR	4009	RSET	400A	RST10	4031	RST18	4034
RST20	4037	RST8	402E	RTOA	0265	SAVE	0462
SCRL	058A	SCRL1	059B	SCROLL	000E	SEC5	0796
SETM	0250	SETR	0208	SINGLE	0001	SOH	004D
SR1	0219	SR2	021F	SSFLG	4004	STACK	4100
STEP	0281	SV1	0489	SV2	049A	SV3	04A4
SVM	01FA	SWITCH	0270	TAPE	0005	TBL1	0230
TBL2	0238	TBL2LN	0018	T1	0445	TIV	4019
TO	04DB	TOP	4001	TOV	4016	TR1	0068
TR2	007A	TR3	0082	TRAP	0049	TRAPX	403D
TRTBL	07B5	UPARR	005E	UPD1	0523	UPDATE	0012
VALUE	0047	VOL	0003	VTOPN	0002	VTPTR	4005
VTV	4010	ZMON	0379				

NO ERRORS